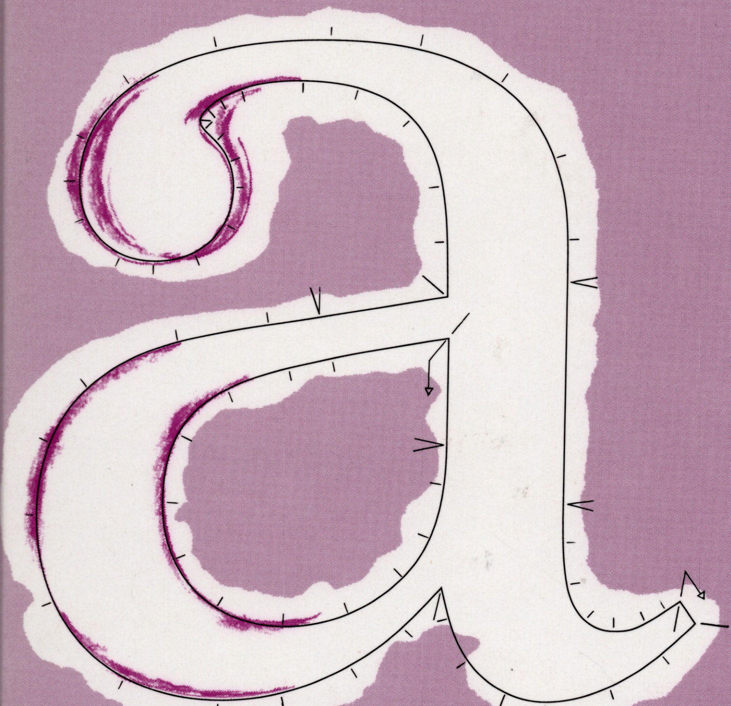
URW

Peter Karow

Digital Typefaces

Description and Formats



Springer-Verlag

Karow · Digital Typefaces

Peter Karow

Digital Typefaces

Description and Formats

Foreword by Hermann Zapf

With 230 Figures

Springer-Verlag
Berlin Heidelberg New York
London Paris Tokyo
Hong Kong Barcelona
Budapest

Peter Karow
URW Software & Type GmbH
Harksheider Straße 102
D-22399 Hamburg
Germany

CR-Klassifikation (1994): I.7.2, J.7

ISBN 3-540-56509-4 Springer-Verlag Berlin Heidelberg New York

Library of Congress Cataloging-in-Publication Data
Karow, Peter 1940- Digital typefaces: description and formats / Peter Karow. p. cm.
Includes bibliographical references and index.
ISBN 3-540-56509-4 (alk. paper). – ISBN 0-387-56509-4 (alk. paper).
1. Type and type-founding – Digital techniques. I. Title
Z250.7.K37 1993 686.2'24 – dc20 93-38414 CIP

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in its current version, and permission for use must always be obtained from Springer-Verlag. Violations are liable for prosecution under the German Copyright Law.

© Springer-Verlag Berlin Heidelberg 1994
Printed in Germany

The use of registered names, trademarks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

Typesetting: URW Software & Type GmbH, Hamburg
Cover design: Konzept & Design, Ilvesheim
33/3140-5 4 3 2 1 0 Printed on acid-free paper.

Summary

This book was written primarily for people who intend or wish to develop new machines for the output of typefaces. It is practical to categorize equipment into three groups for which digital alphabets are required - 1) display devices, 2) typesetting machines and 3) numerically controlled (NC) machines. Until now, development of typefaces has been overly dependent upon the design of the respective machine on which it was to be used. This need not be the case.

Digitization of type should be undertaken in two steps: the preparation of a database using hand-digitization, and the subsequent automatic generation of machine formats by soft-scanning, through the use of a computer-based program.

Digital formats for typefaces are ideally suited to systematic ordering, as are coding techniques. In this volume, various formats are investigated, their properties discussed and relative production requirements analyzed. Appendices provide readers additional information, largely on digital formats for typeface storage introduced by the IKARUS system.

This book was composed in Latino type, developed by Hermann Zapf from his Melior for URW in 1990. Composition was accomplished on a Linotronic 300, as well as on an Agfa 9400 typesetter using PostScript.

Preface

This book was brought out by URW Publishers in 1986 with the title «Digital Formats for Typefaces». It was translated into English in 1987, Japanese in 1989 and French in 1991. Here we present an improved second edition, adapted to the rapid developments. Especially with the chapter on industry standards we have taken into account the fact that desktop publishing and graphic oriented operating systems with computer applications today offer the user more fonts than ever before. Fonts have experienced a «democratisation» and can now be obtained by every user in the desired quantities at reasonable prices.

I am thankful to Mr. Gerhard Rossbach from the Springer Publishers for the encouragement to produce a new edition. I would especially like to thank Jochen Lau for his criticism and collaboration. Essentially, he produced the second edition. Not only did he write new chapters, but also corrected and updated the other chapters. I would also like to thank Mr. Bill Horton, who is responsible for the translation of this edition.

Hamburg, 25th August 1993

Peter Karow

Foreword

This book is aimed at both technicians in electronics as well as users of typefaces in many and varied applications. Much more than an introduction, it touches on a multitude of areas involving digital preparation of typefaces.

In the recent past, typefaces used on visual display terminals (VDTs) were often developed by engineers. Not only did the level of current technology impose serious limitations on the creation of correct letterforms, but similar limits were placed on the joy of experimentation, sometimes a predominant fact in creation of type when done by electronic means. To engineers, aesthetic considerations of letters is too often of secondary importance.

Legibility isn't an old-fashioned concept.

A sizeable number of current digitally generated typefaces are difficult to read, some even ignoring the basic rules of legibility. Let us not forget that legibility is not an outmoded term. Our eyes and reading habits – disregarding optical character recognition requirements – have remained unchanged over centuries.

Consider for a moment the type viewed in modern video communication and television broadcasting. Word images, sometimes using highly unique typefaces, shown only momentarily on the screen, are too often difficult to read by the aged, but also even an average viewer. Letters may be spaced too closely and word images sometimes blended into one another, causing additional reading handicaps. When a sans-serif style «r» is placed too close to an «n», a letter m may be incorrectly perceived. Likewise, «c» and «l» tightly spaced can be mistaken for a «d», and so on. Tightly-spaced typesetting is largely a product of fashion or trend-conscious advertising designers. Almost without exception, every reduction of letter spacing within text leads to an inevitable loss of legibility. Do not forget that rapid recognition of word images remains the primary task of type character reproduction.

Legibility of many visual display terminals could be improved simply by spacing text lines slightly farther apart and letterspacing allowed to revert to a normal constant.

In the days of hot metal composition, due to intrinsic physical properties, it was virtually impossible to print characters too close together. Only upon the arrival of phototypesetting technology and its accompanying electronics could letterspacing be reduced, obtaining the bad habits in vogue today.

The primary purpose of type is to convey information.

As writing is the visual rendition of the spoken word, its primary objective remains the conveyance of meaning to a reader immediately, without difficulties, and free of disturbances to reading flow caused by unnecessary embellishment.

Letters, themselves, have no self-fulfilling purpose. Neither are they a medium for personal self-expression. Their prime task is providing clear differentiation among themselves. This concept applies not only to letters, but to numbers, bearing in mind their importance in contemporary technological societies. Consider how easily the numerals 3, 6 and 9 may be misinterpreted as an 8. In telephone directories – particularly those printed in black on white – distinguishing between individual numbers printed in relatively small sizes rapidly becomes a problem. This and similar problems, leads to an important criteria in new type designs: clearly distinguishable and unmistakable forms.

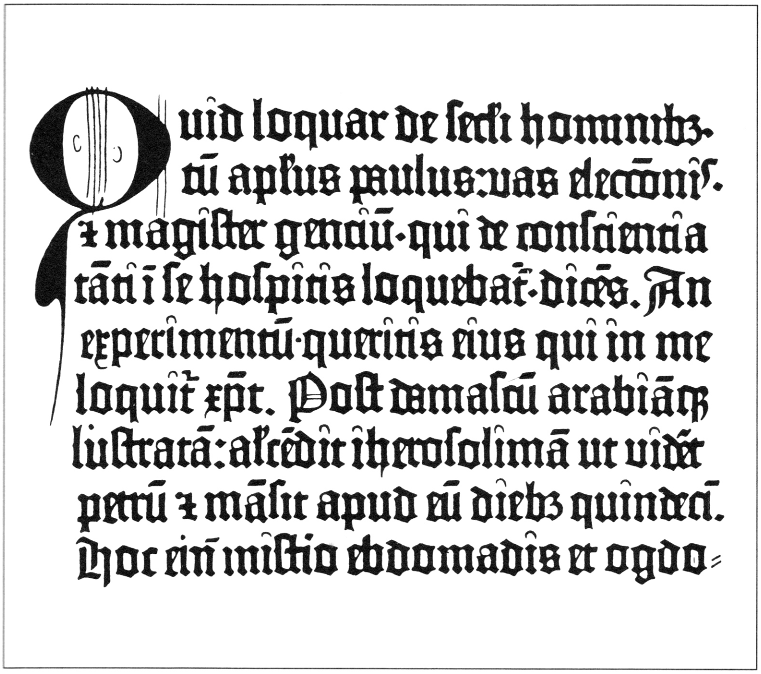
Anything making reading more difficult or time-consuming, or detrimental due to its unusual form, should be avoided in a letter form. When designing a typeface, all these factors, and more, must be carefully weighed from its inception. Of course, drawing characters must take aesthetic requirements into consideration. Eccentricity of any letter forms within a word image may disturb reading flow, with a consequent rapid diminishing of comprehension of text.

This URW handbook provides an introduction to many of the problems encountered in typeface creation and digital typeface preparation. Bear in mind that not everything to do with type is rational and logical. There are purely optical factors as well, the result of lengthy development throughout the history of type, derived from experience and knowledge and definitely not to be ignored. The primary task is no longer to produce incredibly attractive, artistic aphabets as in centuries past. Development and application of new alphabets, as approached by this book, require optimum utilization of technical capabilities now available, in addition to those future methods of type reproduction only dimly seen today.

Our current aim is not to copy old typefaces, Garamond Antiqua from the 16th century or Bodoni from the 18th. We live in an era immersed in technology and electronics. As such, we should attempt to solve current problems by means of contemporary methods.

We should solve today's problems using today's methods.

Most early type fonts stemming from Johannes Gutenberg's time were devised to print on wooden hand presses with predampened paper. Technical limitations of hand-set composition necessarily determined the form of letters.



Quid loquar de ſecli hominibꝫ·
cū apl'us paulus:uas electōis·
⁊ magiſter gencīū·qui de conſciencia
tāti ī ſe hoſpitis loquebat̄·dicēs. An
experimentū queritis eius qui in me
loquit̄ xp̄c. Poſt damaſcū arabiāq;
luſtratā:aſcēdit iheroſolimā ut uidēt
petrū ⁊ māſit apud eū diebꝫ quindecī.
Hoc eīn miſtio ebdomadis et ogdo-

Fig. 1
Text from Gutenberg's-Bible

Infinite possibilities provided by today's electronics should, in like manner, shape the type of our time, removing as much historical baggage as possible. Clear forms should prevail, yet not cold, machine-like ones, which easily lead to unconscious rejection by a reader resulting in an unwillingness to persevere through longer texts.

We are currently entering into an era of new developments in digital application of typographic characters. Previously we were satisfied when text was barely legible on VDTs, while the new trend is to produce innumerable varia-

Typography has previously been limited to two dimensions.

tions from digitally stored basic fonts displaying high legibility. Typography has traditionally limited itself to only two dimensions, but today's technology takes letters into the realm of three-dimensional effects creating images with perspective distortion.

URW's experience with type generation, using the IKARUS system, covers a period exceeding ten years. Peter Karow, using his IKARUS system, was able to develop a new method for mathematical classification of type characters maintaining high aesthetic properties for the first time. The multitude of possibilities provided by modification and interpolation of generated characters, led to new ways of developing entire type families. Results of this benefit today's type users.

Computer-aided mathematical conversion of letter forms will swiftly bring about a creative process permitting the generation of more and more new and attractive solutions using both letters and single word images. The IKARUS system represents the start of this new era in type reproduction also making its presence felt where emerging applications call for increasing numbers of aesthetically pleasing types.

A font on a chip

A font, stored digitally on a chip and having high precision and quality of shape, will aid us in converting the many, often primitive, current visual display devices and their printouts, into pleasant-to-view displays using fonts to which we have become accustomed over the centuries in the spirit of Gutenberg's art of printing. Quality, once taken for granted, should be the standard of the future in the digital type field.

Darmstadt, July 1993

Hermann Zapf

Contents

1. **Introduction** 1

2. **Typeface terminology** 5

3. **Display of typefaces** 19

3.1 Visual display devices 22
3.2 Typesetting machines 30
3.3 NC machines 39

4. **Requirements of typesetting** 45

4.1 Type structure 45
4.2 Form setting 48
4.3 Hierarchical format structure 52

5. **Formats** 57

5.1 Review of formats 58
5.1.1 Bitmaps (bytemaps) 58
5.1.2 Run lengths 63
5.1.3 Open vectors 65
5.1.4 Closed vectors 67
5.1.5 Curved lines 70
5.2 Other concepts 78
5.3 Special coding techniques 81
5.4 Listing of a selection of type manufacturers 85

6. **The URW concept** 89

6.1 IKARUS 89
6.2 Format hierarchy at URW 91
6.3 IK interpolation 94

6.4 Nimbus ... 97
6.4.1 Core-fonts ... 98
6.4.2 Hard scanning ... 100
6.4.3 Scaling ... 103
6.4.4 Selection ... 104
6.4.5 Descriptive elements ... 106
6.4.6 Intelligent font-scaling ... 109
6.4.7 The resolution funnel ... 115

7. The preparation of rasters ... 117

7.1 Soft-scanning ... 118
7.2 Manual corrections ... 127
7.2.1 Corrections on stems ... 127
7.2.2 Corrections on serifs ... 130
7.2.3 Corrections on segments of letters ... 132
7.2.4 Intersection of diagonals ... 132
7.2.5 Stems with slight slopes ... 133
7.2.6 Ink traps ... 133
7.2.7 Tails ... 134
7.3 SC formats for laser printers ... 135

8. Industry Standards ... 139

8.1 Type1 (by Jochen Lau) ... 143
8.2 Intellifont (by Jochen Lau) ... 153
8.3 TrueType Format (by Hartmut Schwarz) ... 161
8.3.1 How the TrueType File is Structured ... 162
8.3.2 TrueType Tables ... 162
8.3.3 Outline Representation ... 164
8.3.4 Scan Converter ... 165
8.3.5 The Instructions ... 169
8.4 F3 Unix (by Jochen Lau) ... 179

9. Conclusions ... 185

9.1 Properties of formats ... 186
9.1.1 Electronic requirements for the generation of control signals ... 188
9.1.2 Storage requirements ... 190
9.2 Requirements to be met by formats ... 192
9.3 Recommendations for manufacturers ... 193
9.4 Time required for producing type ... 194

Appendices

A Type classification 199
B Characters of a typeface 205
C Accents for Latin type 213
D Dingbats 217
E Explanation of optical effects 219
F Short dictionary of type terminology 221
G Data structure for typefaces 235
H Description of the IK format 243
I Description of the BE, CN and QQ format 255
J Description of the II format (by Jochen Lau) 273
K Description of the DI format 299
L Description of the VC format 309
M Description of the BS format 323
N Description of the VE format 339
O Description of the SC format 351
P Description of the SN format 361
Q Description of the BI format 371
R Description of the GS format 381
S Data structure for spacing 389
T Width table 393
U Kerning table 395
V Touching and Overlapping table 401
W Programs for converting IK → DI 403
X Marking and digitizing rules 421
Y References, on typefaces in particular 437
Z Index 441

Introduction

The invention of electronically controlled machines to display text and to output typographic matter has been a key factor in the development of digital type storage. Such machines are divided into three groups according to their application:

- Visual display devices (VDT, graphic VDT, TV)
- Typesetting machines (CRT, laser, microfilm, matrix)
- Numerically Controlled (NC) machines (drawing, cutting, routing, engraving).

There are three types of «writing machines»
VDT= video display terminal
CRT= cathod ray tubes
NC = numerically controlled

Visual display equipment is used to rapidly display text on TV monitors. This purely electronic display with its instantaneous, cost-effective presentation has promulgated poor alphanumeric character quality, compromising legibility, generally not benefitting mankind!

Typesetting machines create text on film, printing plates or paper for a more permanent use or reading at a later time. The hard copy generated is usually of very high quality, excluding for the moment, output from dot-matrix printers still used in data processing.

Cutting, routing, engraving

NC machines draw on paper, cut self-adhesive vinyl materials, cut, rout or engrave wood, metal or plastic. While character quality can range from very poor to unacceptable (such as, letters produced internally by line plotters), using NC machines, some output attains extremely high quality (e.g., flat bed drawing machines cutting materials used for display on hoardings, windows, vehicles, etc.).

Researching digitally stored type we have found that:

- Developments within the three machine groups listed above have been virtually independent of each other. Only rarely do developers become aware of what others are doing.
- Typesetting machine manufacturers have made and continue to make serious efforts to achieve high character quality.

- Integration of development within different fields and machines to produce clean, legible type at reproduction quality has thus far not taken place by involved people.

A majority of development engineers associate «type» only with Latin, Arabic or Chinese symbols employed to reproduce words. Much more must be considered, however, including reproduction of complete textual messages (advertisements, signs, books, newspapers, journals, tables, plans, calculations, drawings). As this is the case, representations of logos, symbols, vignettes, frames and line graphics, as well as elements for generation of line graphics are required in addition to alphanumeric typographic characters.

A font consists of more than an alphabet.

Note: The word «type», used throughout this book, describes all those elements necessary to reproduce a message within the framework of electronic communication, on visual display units, paper or on any other media!

Returning to consideration of equipment, it is true that machine developers are mainly engineers, who, working under pressure of time and the market, must improve their machines, make them faster, more powerful and/or less expensive. Few, if any, give much thought to improving reproduction quality of type. Yet this neglect is difficult to understand as all these devices depend upon writing type to communicate with people. It appears that engineers often extrapolate from their own handwriting, a mode generally serving to communicate at a typing pool or within their limited, circumscribed private sphere. Someone is always handy to decipher writings. This perception leads to an assumption that computer associated type probably doesn't require further thought or enhancement. In turn, this can evolve into an approach where typographic considerations are a necessary evil, its preparation for digital storage taking only a minimum of time and financial expenditure. Two examples illustrate this thinking.

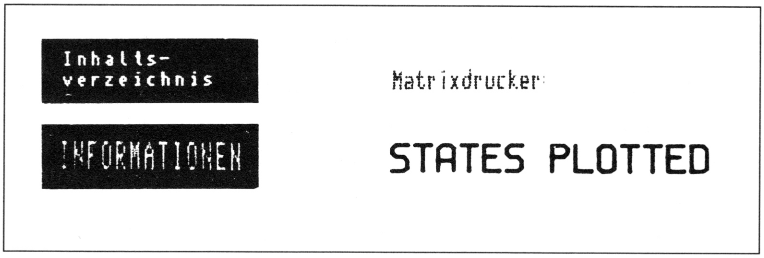


Fig. 2
Examples of poor quality

One inventor of an electronic typesetting machine stated, in reference to type, that development of his typesetting system was complete. A scanner was included in the package so that using supplied equipment, customers were able to produce type for their own use by themselves. The unfortunate fact was, that in the succeeding twenty years, virtually no user had achieved this advertised goal.

Similarly a developer of alphanumeric VDTs, displaying proportional type, spent nearly two years preparing hardware and software for his prototype machine. The amazing thing is, no more than three hours of that time was devoted to proportional type display. He observed, literally, that there was no problem, reflecting his attitude that characters may be classified as either computer-generated or handwritten.

Handwriting and computer type are not the only type species.

In our view, future producers of writing machines would be wise to place greater emphasis on both quality and variety in typographic reproduction! As the three basic groups of computing devices employing type applications are further examined, we observe that all share these properties:

- Type is stored in a manner facilitating conversion from digital form into electric signals as directly as possible; and
- Type has a certain format for data in a certain system environment. For example a PostScript application does not run TrueType fonts.

These two properties have disadvantageous aspects. Within the scope of a particular machine, type is sometimes comprised of a specific digital machine format coupled to a distinct face. Historically, it has been true that digital format generation progresses within parameters of a single machine, slated for introduction into the marketplace as quickly as possible. Few glance beyond production and completion of that current machine. Dynamic successful companies marketing products currently demand a substantial and close link to development of tomorrow's product. However, due to a lack of foresight, type also is developed, digitized, corrected and stored within narrow limitations of specific machine formats, which practice has proven to be expensive! Following such practice, subsequent machine development most often requires duplicate procedures for digitization, correction and storage of type, because an earlier machine's format is often incompatible with that of a newer one.

The second disadvantage noted is that each manufacturer employed its own proprietary machine format, familiar only to the manufacturer. Users attempting to run products from third-party manufacturers on their machines were forced to switch basic type formats resulting in an altered visual appearance, and, in some cases, use of a completely different font altogether.

Changing type and format

These problems can be overcome, when two procedures are put into practice:

1) Digitization of type is executed independent of a specific machine. Digitization must be universal and adaptable to all necessary applications. In other words, all machine type formats must be computer-calculated from a single «mother» format.
2) Machine formats should become standardized with specifications accessible to developers universally. Manufacturers will appreciate this, as type no longer represents a major difficulty to be overcome. Similarly, users will be equally pleased since a change in typesetting machine does not necessarily involve adjusting or adapting new type fonts specific to their new machine.

To demonstrate the possibility of achieving these objectives, this book contains a fairly detailed description and investigation of various equipment classifications, various important machine formats, coding techniques, and finally, demands/problems imposed on type data formats employed in the art of typesetting. Properties of digital type formats are illustrated and their requirements developed.

In the book's final chapter, conclusions are drawn and suggestions made for a future course of action in the manufacture of type data.

Note: This book was typeset with digitally generated type. While reading, be critically aware of text reproduction quality. Aesthetics, we trust you will agree, need not be neglected using this kind of typographic technology.

Chapter 2

Typeface terminology

Terminology current among type designers and producers is used to a limited degree throughout this volume. It is helpful for readers to become familiar with some basic special expressions and terms common to the type world. Let us state, at the outset, we are considering first normally printed letters, black ink on white paper, rather than invisible, abstract characters stored digitally on electronic or magnetic storage media.

Text and display type

Two major headings cover type designs: «text» and «display». Text is type designated as a body of written material as that read in newspapers, books and other informational printed pieces. Display, as its name suggests, is used for titles and headlines, to give special emphasis to an idea. Each of these claims its own special characteristics and style.

With more than 10,000 distinct Latin typefaces currently available, a simplified classification – following ATYPI (Association Typographique Internationale) guidelines – should be useful. (See also appendix A):

Class	Typeface
Roman (Antiqua)	Optima Baskerville **Bodoni** Garamond Palatino Times
Sans serif (Grotesk)	**Futura** Helvetica
Black letter (Textura)	Walbaum Fraktur
Script, Cursive	*Ariston*
Fancy, Free	**Goudy Fancy**

Fig. 3

Many typefaces are supplied in various «fonts» (a font consists of a complete assortment of characters in one size and style). A grouping of typefonts makes up a type «family». Often, though not necessarily, a family consists of four «Roman» (upright) and four «italic» (slanted) versions of progressively weighted characters, each weight given a descriptive name applied by designer or manufacturer. A degree of confusion follows as terms like «Heavy», «Black», «ExtraBold» and «UltraBold» are applied, sometimes within a single family, to describe varying character weights.

light	(mager)
medium	(normal)
demi	(halbfett)
bold	(fett)

Correct designations for typeface weights

International Typeface Corporation (ITC) in New York, one of the most active developers of new typefaces in the world, produces four families a year typically using a clear scheme containing: light (mager), medium (normal), demi (halbfett) and bold (fett) with corresponding italic fonts, a total of 8 distinct variations. Italics of the family are slanted (skewed to the right) usually 12 to 12.5 degrees, on some faces, 15 or more degrees.

A member of a typeface family is properly identified when called, for example, «Optima demi italic». Previous to the electronic age, type sizes were produced individually in hot metal and then set in position for printing. There appears to be no reason why such a tradition should not be continued, even in the age of digital type: we therefore continue the concept as before, whereby special versions were made from a single font in pointsizes of 6, 8, 9, 10, 12, 16, 18, 24 and 36.

Individual alpha-numeric characters may be readily described with familiar denotations. These are: A – Z (caps), a – z (lowercase or miniscules), capital letters at lower case height called «small caps», numerals, punctuation marks (i.e. comma, full stop, question mark, etc.), language variations or accents (ä, ö, ü, œ and ç) with countless other variants also available, such as currency symbols £, $, ¢ and dozens more.

Characters can be raised above their «base lines» (normal positions), e.g., ^{0} (indices, superiors, super-scripts, exponents), and lowered ones, e.g., $_0$ (indices, inferiors or subscripts). Both used together create fractions, e.g., ¼ or $\frac{1}{4}$.

Most European languages require a sizable number of accents, for proper reproduction, while English uses none. A representative list of normal accented characters in various languages is shown:

Czechoslovakian: ÁáČčĎd’ÉéĚěÍíŇňÓóŘřŠšŤt’ÚúŮůÝýŽž
French: ÀàÂâÇçÉéÈèÊêËëÎîÏïÔôÙùÛûÜüÆæŒœ
German: ÄÖÜäöüß
Polish: ĄąĆćĘęŁłŃńÓóŚśZ̀żŹź
Spanish: ÁáÉéÍíÑñÓóÚúÜü
Swedish: ÅåÄäÉéÖö

Examples of accents on Latin characters

Appendix B displays the letters found in a Latin typeface along with their designations. Appendix C displays accents used with Latin letters.

Other terms to consider are those symbols for mathematical formulae written in texts, e.g., the root sign √ or the integral sign ∫. Mathematics cannot, in fact, be written using Latin characters exclusively. Fonts for such setting, as well as its specific symbols, include Greek and Blackletters among its variants. It is interesting to note that the type family AMS-Euler, designed specifically for setting mathematics, was developed by Hermann Zapf and Donald Knuth in the Metafont system (see also, Chapter 5.2) for the American Mathematical Society and Stanford University. To illustrate its diversity, that family contains around 900 characters.

Some fonts have 900 characters.

Other special typesetting situations:

Newspapers, magazines and advertisements all use a varied assortment of line elements to create line borders and other elements to embellish and decorate. Collections of special characters for these tasks mentioned are called «Pi-fonts» consisting of decorative, swash elements, ornamentation, logos and a myriad of specially drawn characters. Typographic devices of this sort attempt to draw and hold a reader’s attention. An example of a Pi-font is the ITC Dingbats series, designed by Hermann Zapf, a portion of which resides in memory of most PostScript printers, a showing appearing in Appendix D.

Ornamentation elements found in Pi-fonts are usually considered vignettes or typographic embellishments, whereas a pictorial representation (a tree, auto or person, etc.) is called a «line graphic».

Characters can be reproduced in «halftones» (with gray tones) or in «patterns», made up of hatched grids, dots, or other graphic effects.

We should make clear that this book is not intended as a course in typography. Descriptions and terminology dealt with thus far merely illustrate some of the complex factors to be considered designing and producing marketable and aesthetically pleasing type. Technicians and programmers engaged with typesetting machines hopefully will be aided by at least a rudimentary knowledge of the basics. Many excellent books detailing typographic matters are widely available (see Appendix Y).

The origins and development of writing is a long and illustrious story. In antiquity, letters or symbols were either chiselled in stone or inscribed or impressed in clay. Examples of this technique, such as Sumearian cuneiform, are to be seen in museums in all parts of the globe. Centuries later, words were scratched or written using a feather stylus (feather quill) dipped in liquid pigment and applied to skins or paper.

In the west, from mid-15th century onward, works began to appear printed from moveable type using some kind of printing machine. Johann Gutenberg in Mainz, Germany is given credit as the first printer to invent and successfully use this new technology to produce books and other largely non-secular matter.

Of interest here is that common to all characters in these early printed works was a largely consistent stroke thickness, both vertical and horizontal. Wedges or small strokes called «serifs» began to appear at tops and bottoms of some letters, hearkening back to incised Roman characters cut in stone centuries earlier.

The capital «D», for example, is described by a single straight and a second curved basic stroke (one straight «stem» and one curved stem). A lower case «n» has no «ascender» nor «descender», whereas «h» has an ascender, «p» a descender. White open spaces in letters (a, A, o, P, q, e, R and others) are called «counters», which accounts for the term «counter punch» as the metal shaft shaped to create the white area of such characters. Characters with a curved stroke diverging from an initial vertical stroke (h, n, m, r) contain upper white «ink traps», requiring special consideration on printing. «O» and «Q» are slightly (approximately 1.5%) taller than «H», causing a noticeable upper «overhang». These same characters

The O overhang should be 1.5% of the cap height.

extend downward, below the «baseline» (body line, type line), creating a «base overhang.» For this reason, curved or circular characters, are usually measurably taller than rectangular characters of the same pointsize to appear optically the same height. «H» is constructed of two vertical «stems» and a horizontal «bar». All three strokes appear equal in thickness when the horizontal stroke is approximately 85% the thickness of the two vertical strokes or stems. This variation becomes more obvious when dealing with «sans-serif» (without wedges or serifs) forms.

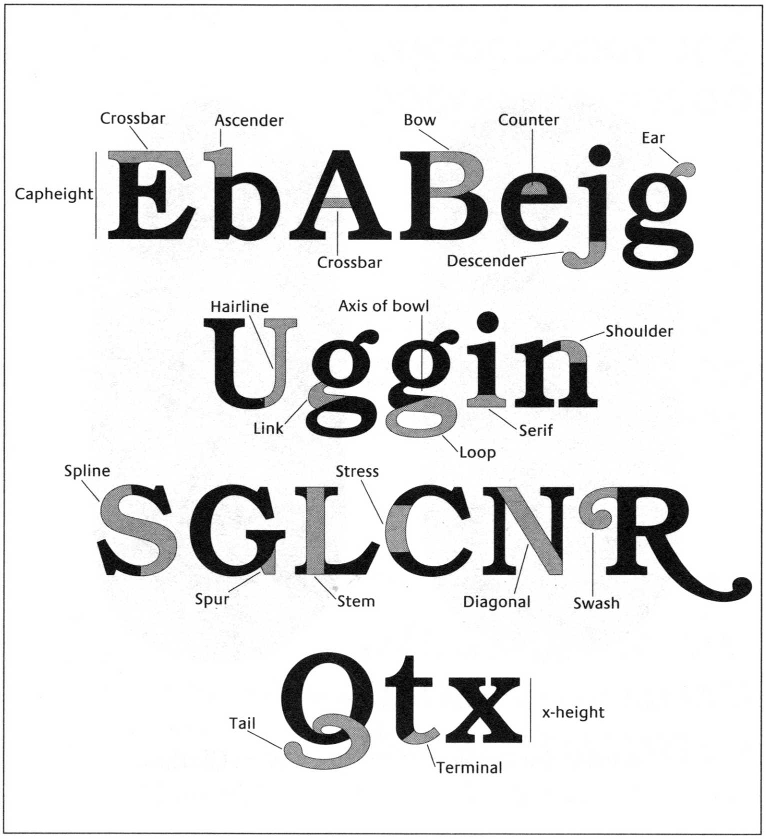

Crossbar
Ascender
Bow
Counter
Ear
Capheight
EbABejg
Crossbar
Descender
Hairline
Axis of bowl
Shoulder
Uggin
Link
Serif
Loop
Spline
Stress
SGLCNR
Spur
Stem
Diagonal
Swash
Tail
Qtx
x-height
Terminal

Fig. 4
Special typographic terms for parts of characters.

Transitions from straight to curved lines, such as in the letter «n», «h» or «m», are known as «tangentials» or «tangents». Such transitions are optically more pleasing if a «clothoid» is applied to the straight stroke. This spiral begins its curvature at zero, gradually increasing along the character's curve. Motorway exits or offramps are frequently clothoid in design to permit

Transitions from straights to curves are like motorway exits.

smoothly entering a curve without wrenching the steering wheel. The degree of rotation of the steering wheel shaft, in actual fact, measures the curvature of a highway turn. Correspondingly, the optical effect of clothoid «tangential transitions» in construction of letters and logotypes facilitates eye movement creating characters easier and more pleasant to read. The significance of tangential distinctions become clear when two semi-circles are placed immediately above and below a standing rectangle. The resulting shape resembles a bone more than a circular stem ending.

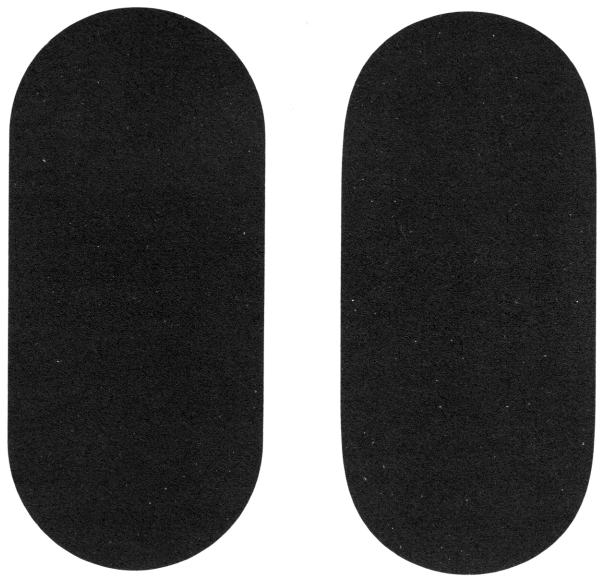


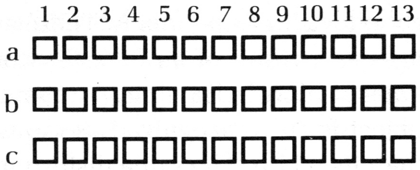
Fig. 5

With semicircles added

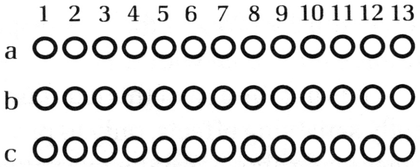
With Clothoid curves

The figure above demonstrates the connection between optical effects and optical illusions. When recognizing or comparing forms and figures, mistakes of recognition do occur.
As an exercise, some basic questions are posed to the reader:

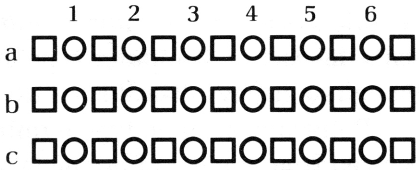
1. Which rectangle appears to be a square? Please cross.


1 2 3 4 5 6 7 8 9 10 11 12 13
a
b
c

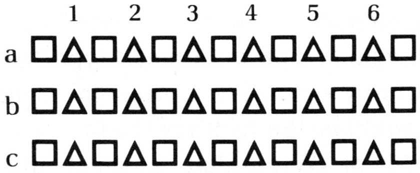
2. Which ellipse appears to be a circle? Please cross.


1 2 3 4 5 6 7 8 9 10 11 12 13
a
b
c

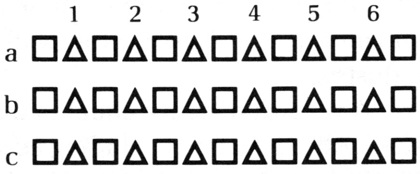
3. Which circle appears to be as high as the squares? Please cross.



4. Which triangle appears to be as wide as the squares? Please cross.



5. Which triangle appears to be as high as the squares? Please cross.



Make a cross on the figures you think come closest to the solution. You will find the results of our small statistical survey in Appendix E.

Fig. 6

You will find a compilation of explanations and details of optical effects in Appendix E. These are based on statistical averages as a result of an investigation involving 130 test candidates at URW in Hamburg. In summary, we found that a square object is only recognized as «square» when drawn 1% taller than its width. The same distortion was seen using circles, although curved objects begin to appear as large as adjacent squares only when their diameter extends 3% beyond the width of square objects, such as «H» or «M».

Squares should be 1% higher than wide.

In the case of triangles, (A and V), height should be 3% taller, width 5% broader than corresponding square-sided widths to appear of equal width.

Exaggerate the apex

It is not the aim of this exercise to detail production of optically satisfying effects, only to provide guidelines. The percentages given above are «mean» values. Each particular typeface, as might be expected, has its own peculiar effects not to mention its own individual «excesses». It was learned, also, that such subtle typographic effects are mainly manifested by experienced type designers with a well-trained eye. Future type technicians might bear this fact in mind. It is hoped we have seen the last of computer typefaces completed in 3 hours, though such may not be the reality.

Returning again to type terminology, we find more terms derived from the era of hot metal composition. Each digitally stored character contains a mathematically derived «reference point», known in Cartesian geometry as the coordinate «zero point» or «origin». This point has come to be located at the lower-left corner of a rectangular box («bounding box») into which all characters of a typeface fit, even the widest. This rectangle (a square in metal type) was called an «em square» or simply «em». The horizontal measurement of the box or em would be the width of the broadest character in a font, usually «M» or «W», which likely accounts for its name. Splitting the em equally in two creates two «ens». Hot metal compositors used the appelation «mutt» and «nutt» in place of «em» and «en», a practice which seems to be disappearing with the advent of photo-electronic typesetting devices. The actual em and en spaces remain currently in use within the graphic arts industry.

A vertical distance equal to the em width is called «bodysize». All characters, including accents and specials, normally fit within the height of this measurement.

The breadth of a character is called its «width», (measured in digital units), often quite small for the letter «i» or «l» yet often equal to bodysize in the case of «m» or «W».

Bodysize must also be tall enough to include accents, ascenders and descenders. A commonly used division is:

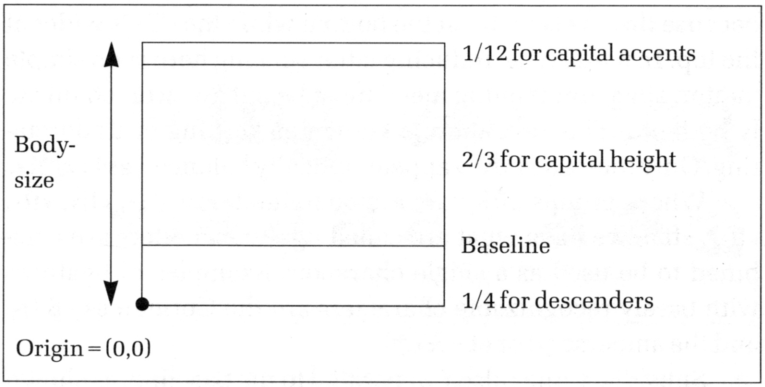

1/12 for capital accents
Body-size
2/3 for capital height
Baseline
1/4 for descenders
Origin = (0,0)

Fig. 7
Em square divisions

Later in section 9.2, we propose a new em square, not to confuse or irritate friends, but to make life easier in the future.

Individual letters rarely occupy the width alotted them; they have a left and right side bearing (the white space to the left and right of the letter).

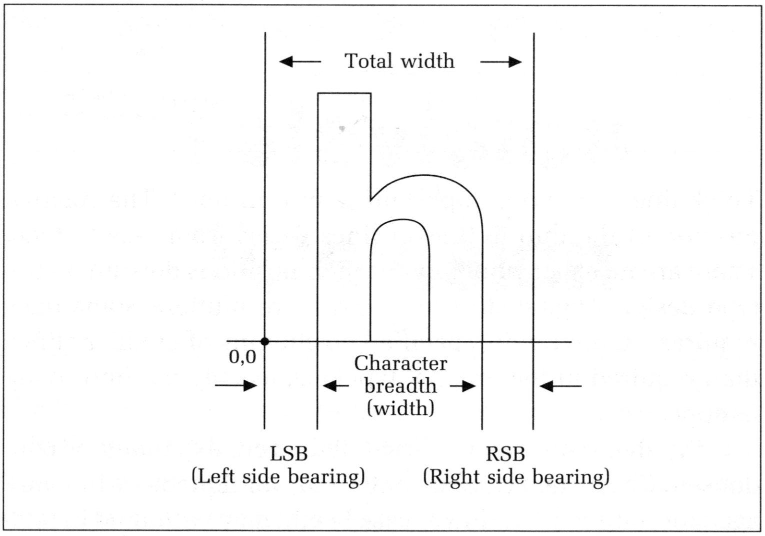

Total width
0,0
Character breadth (width)
LSB (Left side bearing)
RSB (Right side bearing)

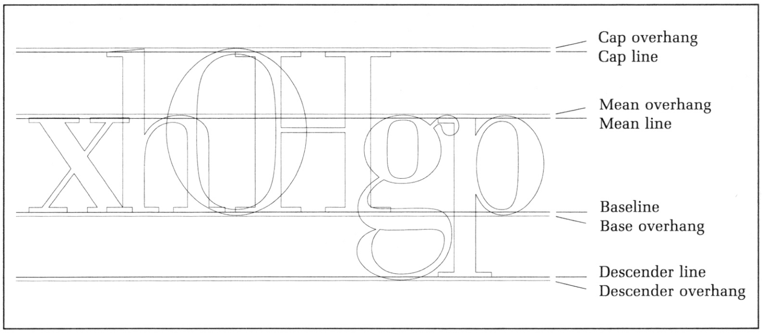
Fig. 8
The total width is composed of the character breadth (width) and left and right side bearings.

The left and right side bearings are intended to allow a letter to appear optically balanced placed next to any other character of its own typeface. Of course, in practice, this is a compromise, as spacing between letters depends upon the form preceding and following it. Consider, for example, the «A» and two possible following neighbors «B» and «T». In hot metal composition we would have the following images, «AB» and «AT». In the case of «AT» the «T» can be moved closer to the «A», because the «A» is wider at the bottom while the «T» is wider at the top. In metal type, reducing letter spacing here is no simple matter, since intervening metal must be cut away mechanically by hand. This operation is known as kerning or undercutting. Only then does «AT» appear optically balanced as is «AB».

Where groups of letters are combined, say «ff», «fi», «fl», «ffi», «ffl», we have what are called ligatures. Letters are combined to be used as a single character. Examples of ligatures with barely recognizable characters are the German ss - ß (ſs) and the ampersand or et – & (&).

Subsidiary lines also run parallel to the baseline, as shown in the following diagram:

Fig. 9
All font lines at a glance



Thick lines are more important than thin ones. The relative position of the thin to thicker lines varies from face to face. There are no exact guidelines as their positions depend on the type design. Digital storage at coarse resolutions sometimes requires a more rigidly specified positioning of auxiliary lines than required in the original typeface, usually not improving its appearance.

Typeface modification

Typefaces may be modified, italicized, expanded or condensed. Characters can be contoured, «antiqued» (artificially aged) or rounded. The lower case height may be altered in relation to cap height, whole words blended together, or letters given various shading, in general, creating entirely new letter forms.

Using simple geometric operations characters may be enlarged, condensed, rotated and distorted. Envisaging transformations such as these is no problem for technicians, but unfortunately there exists a body of devices incapable of these transformations at all. This includes many VDTs.

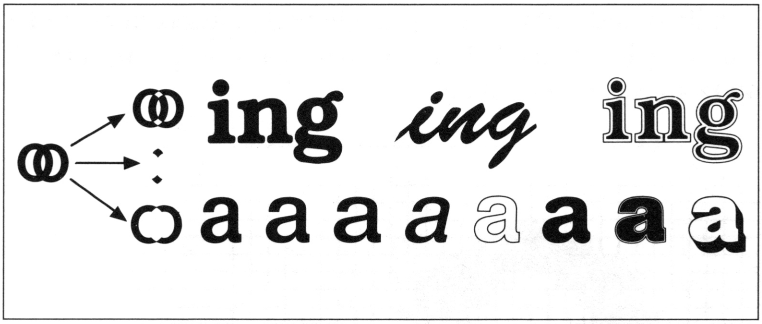


Fig. 10
Examples of character modification

Currently, still many typewriters display «`monospaced`», i.e. all characters occupying the same horizontal space (also termed «non-proportional»).

Typesetting machines use proportional widths, a significant advantage. Setting is further improved when not only the right side bearing is adjusted to left side bearing of the following letter, but when space between any two letters is also adjusted specifically for that pair of letters, i.e. using the technique of digital kerning. Consider for instance:

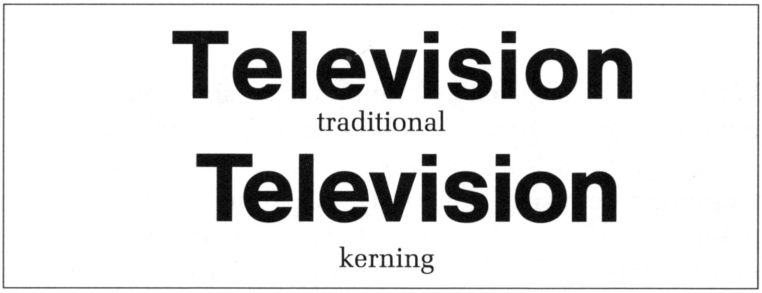
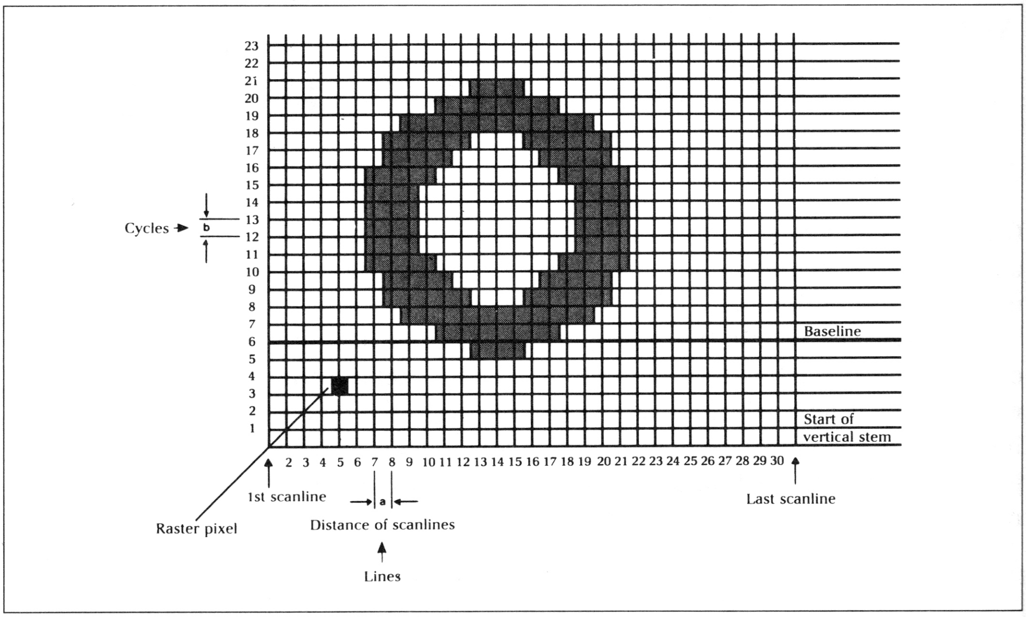


Fig. 11
Kerning improves traditional setting.

Finally, there are a few special terms associated specifically with the digitization of type. All printing devices write with a certain line resolution, specified in either «lines per inch», «lines per millimeter» or «dots per inch». This means that the black portions of letters are machine generated (e.g., typesetting machine) so that black lines of a certain thickness are drawn, stroke by stroke.

Another limitation of digital technology pertains to the line itself; the writing is only switched on or off in certain electronically specified periods or cycles. Accordingly white or black sections of a line consist of fixed length elements (pixels). The letters appear as lines and dots on a horizontal and vertical raster (grid). In a 400 × 400 raster an em square consists of 400 scanlines and 400 pixels per line in both x (horizontal) and Y (vertical) directions. This may also be described as an em with 400 × 400 raster resolution.

Fig. 12
Soft scanning of characters with black pixels


23
22
21
20
19
18
17
16
15
14
13
12
11
10
9
8
7
6
5
4
3
2
1
Cycles
b
Baseline
Start of vertical stem
2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30
1st scanline
a
Distance of scanlines
Lines
Last scanline
Raster pixel

We at URW believe that describing a letter according to its em is the most effective way of visualizing digitization of type.

Didot
1 p=0.375 mm

One Didot point (1 p) is the generally accepted unit in measuring pointsize. Previously, according to Didot, $62\,{}^{2}/_{3}$ points = 23.565 mm. In Europe today, one normally uses the new Didot point with dimensions 1 p = 0.375 mm. In the U.S., standard practice is using the pica point, where 1 pt = 0.351 mm. One inch (25.4 mm) has approximately 72 points (precisely 72.27 points).

Pica
1 pt=0.351 mm

Some legislators would like to see a new dimensioning of capital letter heights (H-height) in mm, rather than in bodysizes in points. At this time, typesetting machine manufacturers give measurements in both systems, but as the world of printing is quite conservative, the old units are frequently still in use.

Text in newspapers usually appear in 9 p ≅ 3.4 mm bodysize = 2.25 mm cap height. Occasionally, 10 point type is used.

Typewriters often have 9.5 p ≅ 3.6 mm bodysize, 2.4 mm cap height. Some have 10 pt = 3.51 mm bodysize. Considering the 9 p em as standard, we then have:

Writing resolution in		Raster for
l/mm	l/inch	the 9 p em
7	180	24 × 24
9,4	240	32 × 32
11,8	300	39 × 39
15,7	400	54 × 54
32	813	108 × 108
64	1626	216 × 216
96	2438	324 × 324

When a 9 p Didot em is placed next to a 9 pt pica em, magnified eight times, a distinct difference in bodysize can be recognized (approximately 7%) yet the H-Height remains the same. The pica em of most manufacturers allows less room for descenders.

Didot = Pica + 7%

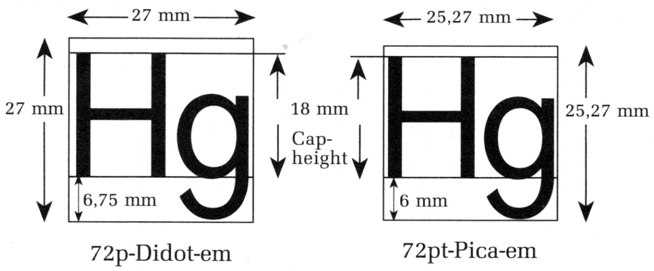


Fig. 13
The cap heights are the same size in Didot em and pica em.

As bodysize determines line width of text, setting with pica points allows the presentation of English texts with the same cap height, but small line widths because no accents collide with descenders of the preceding line. There is, in fact, a simple conversion method from Didot points (new typographic points) to mm: The ratio of cap height to bodysize is 2:3.

Similarly, 1 p × 2/3 = 0.375 × 2/3 = 0.25 mm. Thus, one can derive cap height in mm by dividing the bodysize (in points) by 4.

X **mm** *VH*=**4***X* **p**

4 p bodysize = 1 mm cap height

The same is true for pica points as shown above.

Low
&
high resolution

Rasters of 50 × 50 and smaller are known as «low resolution» while rasters in the 100 × 100 range and more are known as «high resolution».

Any typographer could greatly expand this chapter with much greater detail. However, we conclude with the aforementioned traditional terms.

See Appendix F for a short listing of English, French and German terms for and about typefaces.

Chapter 3

Display of typefaces

Because the primary interest in this book involves type using digital formats it will be helpful to include all devices displaying typographic characters. As previously mentioned, those machines can be grouped into three categories; visual display terminals, typesetting machines and numerically controlled (NC) machines.

All machines considered in the three categories hold in common two characteristics when creating a character: 1) The written character has been produced from data digitally stored somewhere within the machine or system, either electrically or magnetically. 2) Each character's data is assigned a binary address, thereby permitting electronic selection of any stored character, accessed in any sequence.

How does digital control work?

The first step in the writing process is reading character information and temporarily storing that digital data in fast-access memory. Next, the processing unit configures the image data digitally to produce the appropriate vertical and/or horizontal black line segments. Such a digital line consists, for example, of an alternating series of zeros and ones, known as bits (see sketch). From this digital sequence it is possible for the processor to generate an electronic signal which drives the writing mechanism, thus producing, out of the following bit line:

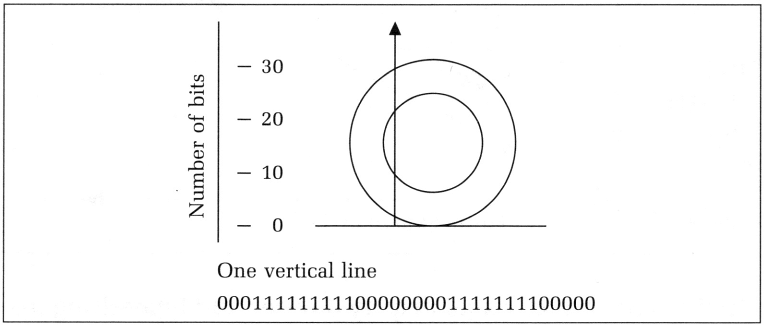


Fig. 14
Black lines are scanned and stored as a sequence of bits.

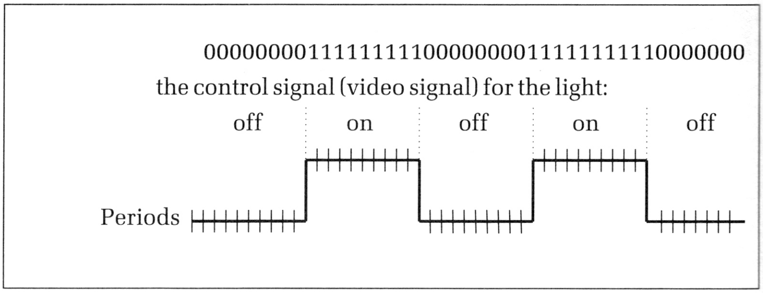

00000000111111111000000001111111110000000
the control signal (video signal) for the light:
off
on
off
on
off
Periods

Fig. 15
Bits are converted into time periods of identical length.

The beam of any particular machine has a specific thickness, determining line width and separation.

Plotter and oscillograph control

NC machines and also some graphic visual displays prepare their control signals in a different manner. Their writing mechanisms direct pens, knives, routers, etc. one point to another in both a horizontal and vertical direction simultaneously. To accomplish this task, stored digital data is used to generate two parallel control signals, one for X or horizontal movement, the other for simultaneous vertical or Y movement. The two controlling signals each contain the desired number of unit steps of movement to be taken from point to point. Adding path segments in X and Y directions simultaneously produces a straight linear movement. These straight line segments are known as «vectors».

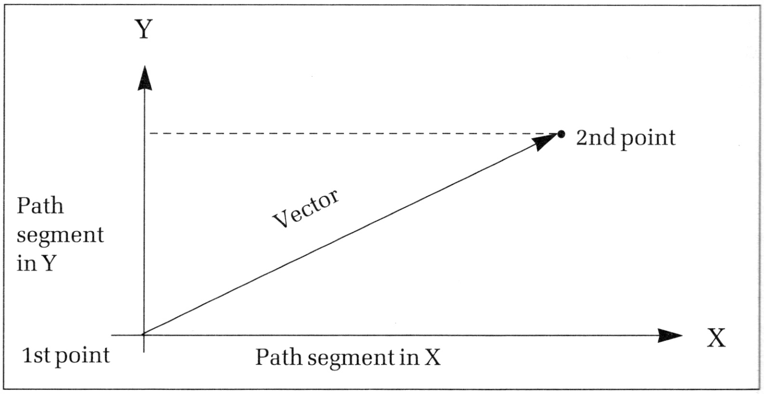
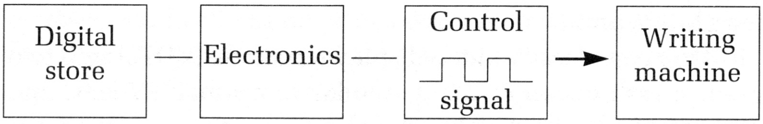

Y
2nd point
Vector
Path segment in Y
1st point
Path segment in X
X

Fig. 16
NC machines and oscillographs execute movements in the X and Y directions simultaneously.

On the other hand, visual display devices and typesetting machines generate lines parallel or perpendicular to the writing direction, moving forward line by line across the screen or film.

No matter what technique is used, machines in all their groups can describe a surface by generating the appropriate signals in the x and y directions.

The point of greatest interest to us is what the machines write and how the results appear. To do this we have generally abbreviated the electronic mode of function in each machine as follows:


Digital store
Electronics
Control signal
Writing machine

A simplified illustration shows how the control signals and light source interact in a laser typesetter (see below).

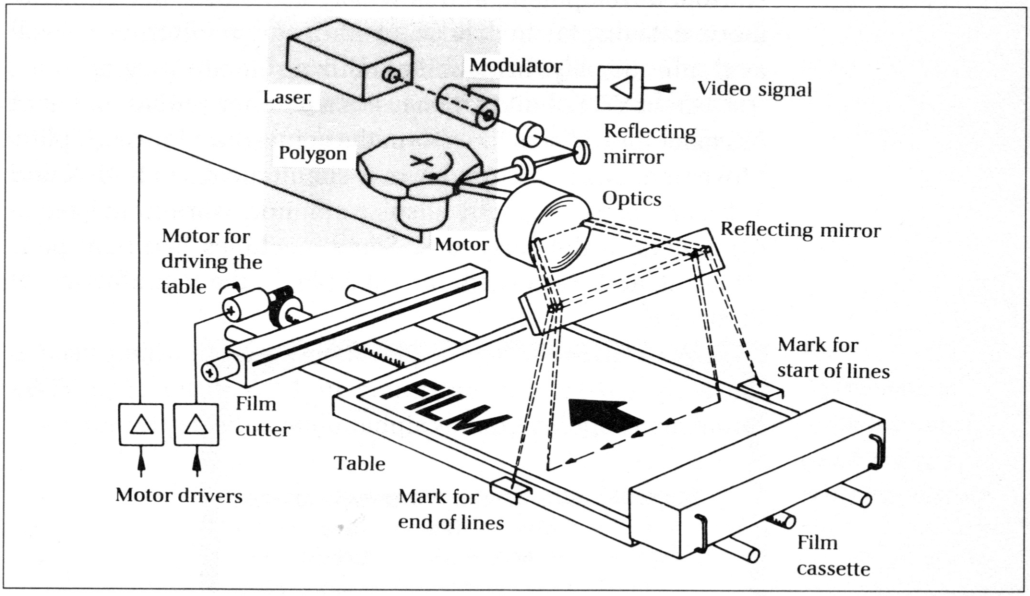

Laser
Modulator
Video signal
Reflecting mirror
Polygon
Optics
Reflecting mirror
Motor
Motor for driving the table
Mark for start of lines
FILM
Film cutter
Table
Motor drivers
Mark for end of lines
Film cassette

Fig. 17
Example:
Digiset control

3.1 Visual display devices

In such devices, control signals are utilized to direct the electron beam of a cathode ray tube (CRT). The electron beam impinges on a phosphorescent coating on the screen, generating a bright point.

Raster scan VDTs versus direct VDTs

One can classify video display terminals (VDTs) into two groups: raster scan displays, similar to normal TV sets, and direct VDTs, similar to laboratory oscilloscopes. The first generate an image electronically on a screen, line by line horizontally, and repeating this process faster than a normal eye can perceive. This system appeared in 1970 VDTs as a computer console development, and is widely used today for alphanumeric data display in data acquisition and monitoring, as well as graphic displays in machines utilizing bitmap imaging.

Direct VDTs build an image in a manner similar to that of NC machines. Visual lines form the actual drawing or graphic. Movement of an electron beam is simultaneous in both X and Y directions. Such graphic display monitors were connected to computers in the early '60s. Some machines had light pens attached, and were used, for example, in the manufacture of PC boards.

Direct VDTs (storage display) became more widely used in the early 1970's, and beginning in 1975 raster scan VDTs found many applications as alphanumeric display terminals.

First generation display terminals

Display terminals referred to as first generation have a light beam traversing line by line from left to right of the screen. Neighboring raster points join to form one line, whose length corresponds to the number of points the electron beam was switched on. In the digital display of text characters, relatively small rasters for an em are used (e.g., 8 × 10 units).

Raster scan VDT

In the following series of pictures, the pairs of images have the following relationship: Photo 1 shows a full frontal view of the graphic display terminal and the page composed or the paper drawing (depending upon device group). This produces different reductions.

Photo 2 is a 5:1 enlargement of four square centimeters of the original surface on which letters are displayed. The first pair, below, shows the image of letters on a normal display unit. The resolution is approximately 2 lines/mm or 50 lines/ inch.

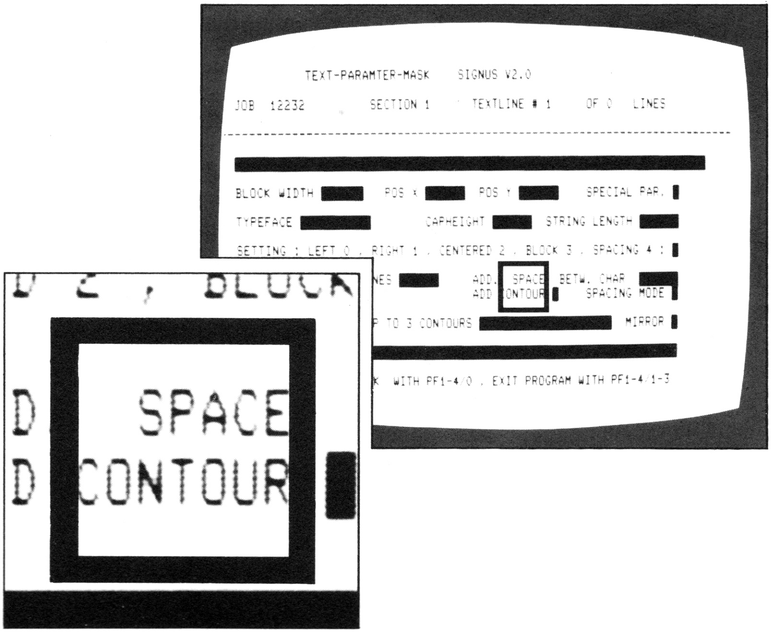

TEXT-PARAMTER-MASK SIGNUS V2.0
JOB 12232 SECTION 1 TEXTLINE # 1 OF 0 LINES
BLOCK WIDTH
POS X
POS Y
SPECIAL PAR.
TYPEFACE
CAPHEIGHT
STRING LENGTH
SETTING : LEFT 0 , RIGHT 1 , CENTERED 2 , BLOCK 3 , SPACING 4 :
NES
ADD. SPACE BETW. CHAR
ADD CONTOUR
SPACING MODE
P TO 3 CONTOURS
MIRROR
K WITH PF1-4/0 , EXIT PROGRAM WITH PF1-4/1-3
BLOCK
D SPACE
D CONTOUR

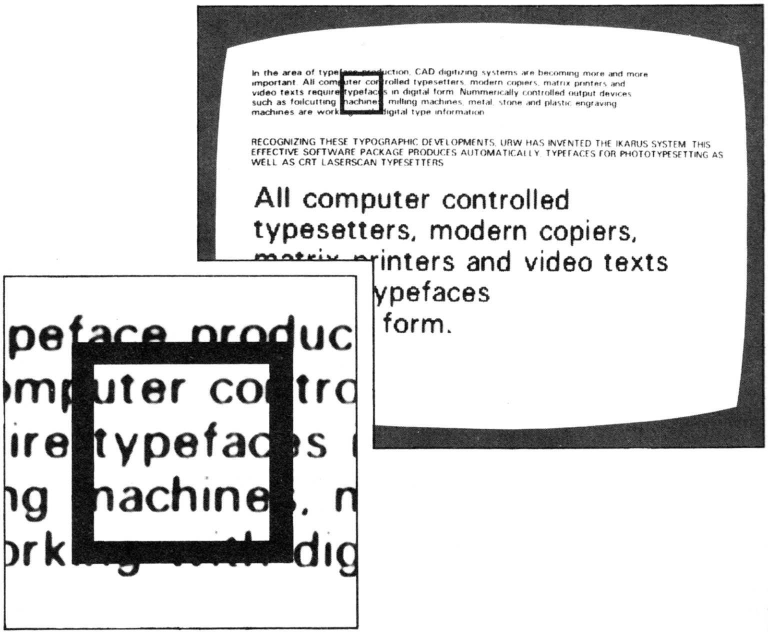
Fig. 18
Alphanumeric display terminal with black and white type display

Second generation display terminal

Gray representation

Display terminals bearing this description operate essentially as do first generation alphanumeric raster display terminals, except that they store 16 × 16 halfbytes per letter. As they use more lines, they need not necessarily write monospaced characters. Such improvements require processing electronics to work 16 times faster than before with a resulting improvement in resolution of approximately 4 lines/mm or 100 lines/inch.

Fig. 19
Character-based terminal with type displayed in gray



In the area of typeface production, CAD digitizing systems are becoming more and more important. All computer controlled typesetters, modern copiers, matrix printers and video texts require typefaces in digital form. Nummerically controlled output devices such as foilcutting machines, milling machines, metal, stone and plastic engraving machines are work[illegible] digital type information.

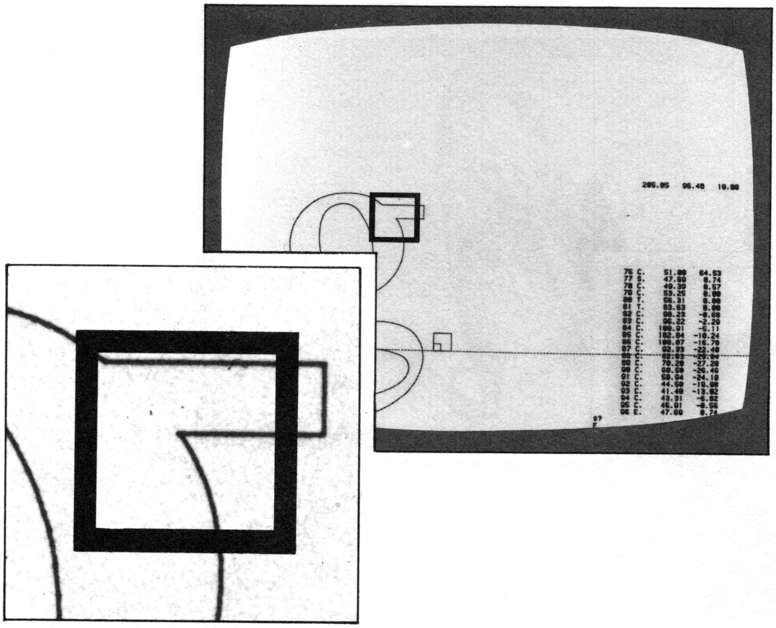
RECOGNIZING THESE TYPOGRAPHIC DEVELOPMENTS, URW HAS INVENTED THE IKARUS SYSTEM. THIS EFFECTIVE SOFTWARE PACKAGE PRODUCES AUTOMATICALLY TYPEFACES FOR PHOTOTYPESETTING AS WELL AS CRT LASERSCAN TYPESETTERS.

All computer controlled
typesetters, modern copiers,
matrix printers and video texts
ypefaces
form.

First generation graphic display terminals

Graphic display terminals of this designation (direct VDTs or storage displays) have very rapid line generators for X and Y control of the electron beam. Data to be displayed is stored graphically as vectors with each vector linking two points (knots). Resolution may be up to 10 lines/mm or 250 lines/inch, the width of the electronic beam is, however, only 4 lines/mm or 100 lines/inch.

Direct VDT



```
                 205.05   96.40   10.00




           76 C.     51.00   64.53
           77 S.     47.60    0.74
           78 C.     49.30    0.57
           79 C.     53.25    0.00
           80 T.     56.31    0.00
           81 T.     83.63    0.00
           82 C.     90.23   -0.60
           83 C.     96.22   -2.29
           84 C.    100.01   -5.11
           85 C.    102.04  -10.24
           86 C.    100.07  -16.79
           87 C.     92.83  -22.40
           88 C.     82.83  -25.04
           89 C.     70.20  -27.20
           90 C.     60.50  -26.46
           91 C.     50.04  -24.10
           92 C.     44.50  -10.98
           93 C.     41.40  -13.82
           94 C.     43.31   -6.82
           95 C.     46.01   -0.56
           96 E.     47.60    0.74
     97
     F
```

Fig. 20
Graphic display terminal operating on oscillograph principle

Second generation graphic display terminals

Raster graphics

This class of graphic display, the raster display terminal, generates its entire screen display from a corresponding bitmap of 800 × 600 bits (600 horizontal lines each containing 800 points). Displays such as these widely replaced storage display terminals. Their line imaging is slightly less, but there is little difficulty filling a screen with black or white. Resolution ranges up to 4 lines/mm or 100 lines/inch.

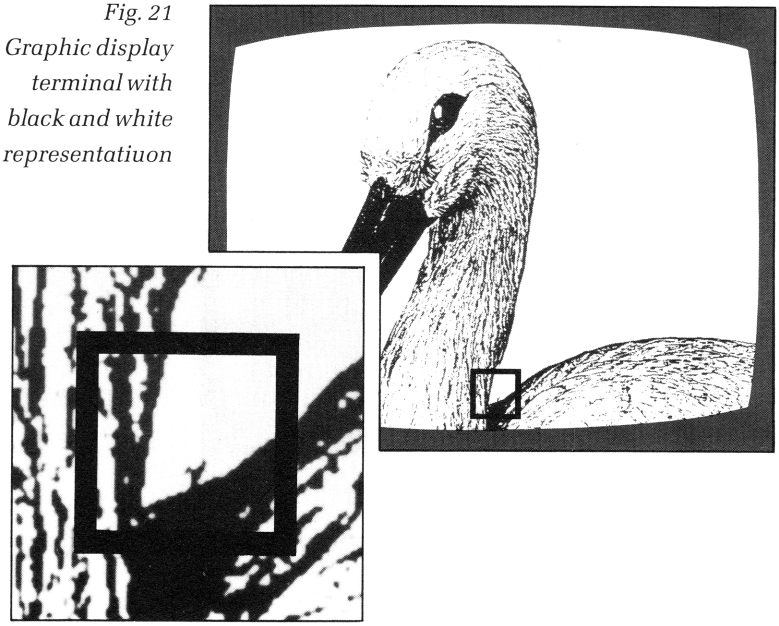
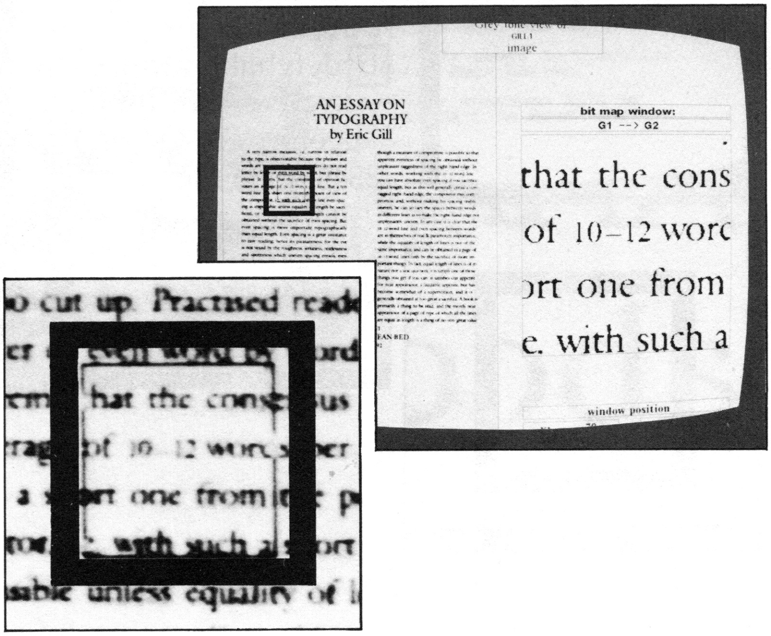


Fig. 21
Graphic display terminal with black and white representatiuon

Second generation graphic display terminals with halftone display

A fairly recent development is the raster display terminal which displays not only black and white points but also numerous (up to 256) intermediate gray tones. The image of the pixels (picture elements) is read from a quarterbyte-, halfbyte- or bytemap with corresponding four (quarterbyte), 16 (halfbyte) or 256 (byte) gray tones. Lines are well imaged and resolution is normally up to 4 lines/mm or 100 lines/inch.

Graphics with gray tones



Grey tone view of
GILL1
image

AN ESSAY ON
TYPOGRAPHY
by Eric Gill

bit map window:
G1 --> G2

that the cons
of 10–12 worc
ort one from
e. with such a

window position

Fig. 22
Graphic display terminal with gray level display

Television

Analogue display

The type seen on normal TV screen usually comes from two sources, either a TV camera focused directly on two-dimensional typographic copy:

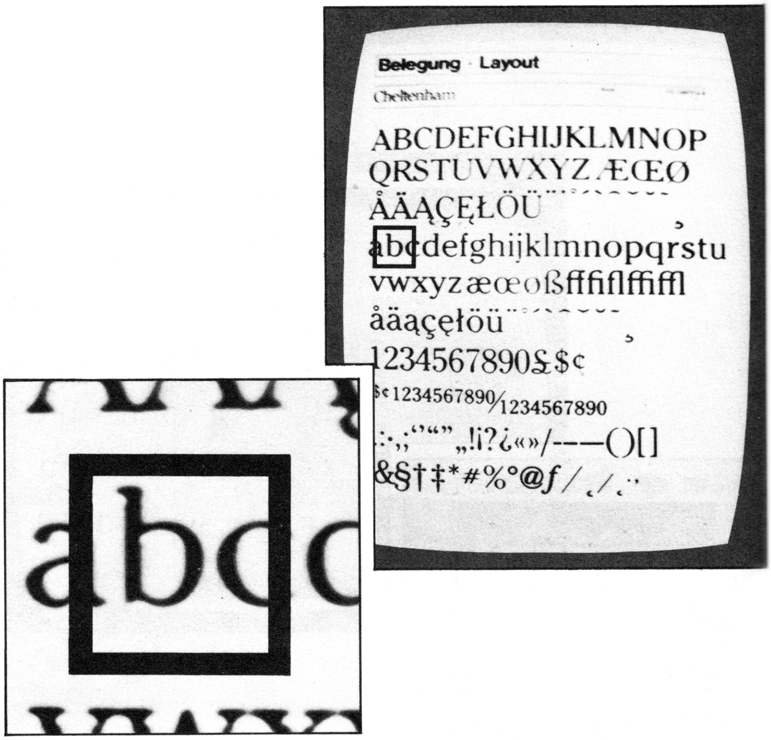
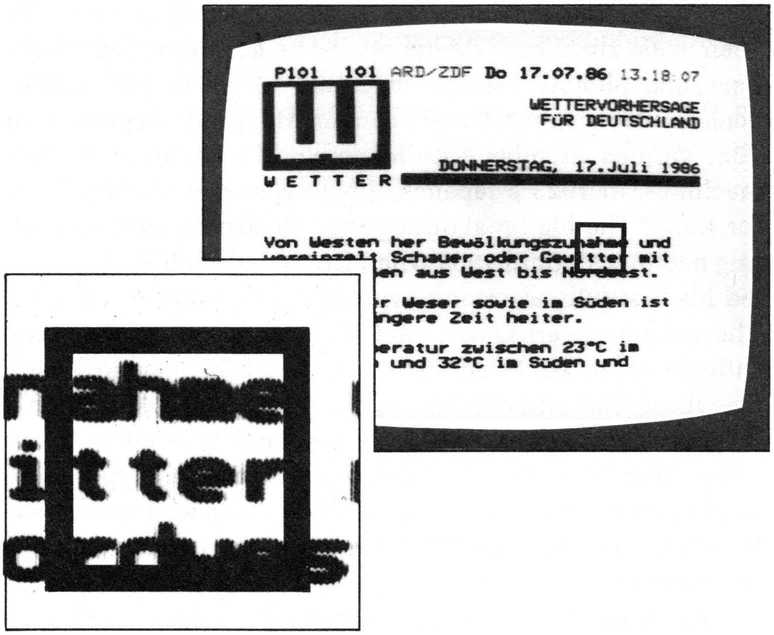

Belegung Layout
Cheltenham
ABCDEFGHIJKLMNOP
QRSTUVWXYZ ÆŒØ
ÅÄĄÇĘŁÖÜ
abcdefghijklmnopqrstu
vwxyz æœ oßfffiflffiffl
åäąçęłöü
1234567890£$¢
$¢1234567890/1234567890
:·;;‘’“” „!¡?¿«»/–––()[]
&§†‡*#%°@f /
abc

Fig. 23
TV representation of type, filmed from a display board

or the type is produced using a typesetter-like processor and mixed into the analog video signal. Standard resolutions are upward of approximately 2 lines/mm or 50 lines/inch.

Digital representation



P101 101 ARD/ZDF Do 17.07.86 13.18:07

WETTERVORHERSAGE
FÜR DEUTSCHLAND

DONNERSTAG, 17.Juli 1986

WETTER

Von Westen her Bewölkungszunahme und
vereinzelt Schauer oder Gewitter mit
en aus West bis Nordwest.

r Weser sowie im Süden ist
ngere Zeit heiter.

eratur zwischen 23°C im
und 32°C im Süden und

Fig. 24
TV display of videotext

3.2 Typesetting machines

Typesetting machines, as referred to in this book, include: matrix printers, electrostatic printers, laser printers, microfilm devices, CRT machines, laser film recorders and other less widely used printing machines. The term «typesetting machine» is traditionally used in a more limited context.

Typesetting machines for hot metal composition have been in use since 1886 (Mergenthaler), e.g., Linotype or Monotype, and phototypesetting machines for Latin letters have been available since 1949 (Higonnet, Moyroud). Beginning in 1922 various attempts were made world-wide on innovative machines; in 1923 a Japanese phototypesetter was invented for Kanji. The big breakthrough for electronic phototypesetting had to wait until 1965 when Dr. -Ing. Rudolf Hell presented his first fully electronic typesetting machine, the Digiset. These machines are known as «CRT machines» as they employ cathode-ray tubes. From 1975 onwards, CRT machines took over the market and virtually replaced hot metal composition.

Dr.-Ing. Rudolf Hell invented the electronic typesetting machine.

Between 1974 and 1978 the first laser typesetters (Linocomp from Linotype and Lasercomp from Monotype) were produced and marketed. In recent times displacement of earlier phototypesetting and CRT machines by laser machines has come to pass.

All of these machines are described by the overall term «typesetting machine». However, as mentioned earlier, we also include other devices such as matrix printers, electrostatic printers, laser printers and micro filming devices under this term. As technology changes, resolution increases, distinctions between various machines has become somewhat blurred.

Matrix (or dot-matrix) printers have been available since 1970 and have found widespread use as inexpensive alternatives to speedy line printers (which use metal type printing characters) for printing the output of electronic data processing. Over approximately the same period, the use of electrostatic printers has become more widespread in the field of computer aided design (CAD) for generation of drawings including text. Such printers compete to some extent with plotters and drawing machines.

Laser printers have been successfully marketed by XEROX, Canon, etc. from 1980 to the present. Their first real breakthrough into larger market segments is as recent as 1984. The ubiquitous Canon printing engine from Japan was one of the leading elements to account for this surge. These printers, connected to personal computers (PCs) is undoubtedly the office typewriter of today, and is used in huge quantities in the field of desk top publishing.

Microfilming devices represent a special variant of typesetting machine. They produce film sheets to be viewed with enlarging devices. Increasing capacities of computers and data networks likely marks the demise of such devices.

Overall, we think that the only devices to survive into the future are high resolution dot matrix printers (24 to 48 pin), laser printers and laser film recorders (laser typesetters).

Laser technology has a future.

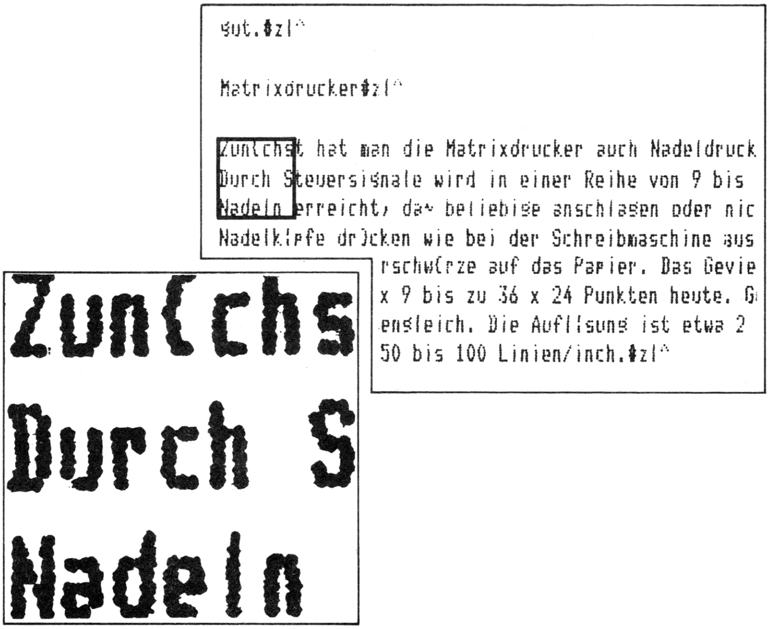
Matrix printers

These printers utilize a printing head consisting of a grouping or matrix of from 9 to 48 pins, oriented perpendicular to the writing direction and actuated by a control signal. The stricken pins force ink from an ink ribbon onto the paper, similar to a typewriter. The em consists of from 7 × 9 up to 48 × 36 points or dots on current machines. Monospace typesetting is the norm in these machines with resolutions ranging from approx. 2 to 7 lines/mm or 50 to 180 lines/inch, some even reaching 360 lines/inch.

Still only monospace setting?

When preparing the first German edition of this book in 1984/85, text copy was available only in the form of lists or galleys printed out with a matrix printer. On reading these proofs, many orthographic errors were overlooked, later to be found when the final proofs were read (produced by a Digiset).

Fig. 25
Many matrix printers have poor quality type reproduction.


```
gut.#zl^

Matrixdrucker#zl^

Zun(chst hat man die Matrixdrucker auch Nadeldruck
Durch Steuersignale wird in einer Reihe von 9 bis
Nadeln erreicht, da~ beliebige anschlagen oder nic
Nadelk!pfe dr)cken wie bei der Schreibmaschine aus
               rschw(rze auf das Papier. Das Gevie
               x 9 bis zu 36 x 24 Punkten heute. G
               ensgleich. Die Aufl!sung ist etwa 2
               50 bis 100 Linien/inch.#zl^
```
Zun{chs
Durch S
Nadeln

Electrostatic printers

Electrostatic printers or electrostatic plotters use control signals to produce electrostatic charges on lines running parallel to the writing direction, which, in turn, attract ink particles. These ink particles, absorbed by the paper, produce the image. Typical resolution values are 8 lines/mm or 200 lines/inch, with newer machines capable of 16 lines/mm or 400 lines/inch.

200 lpi

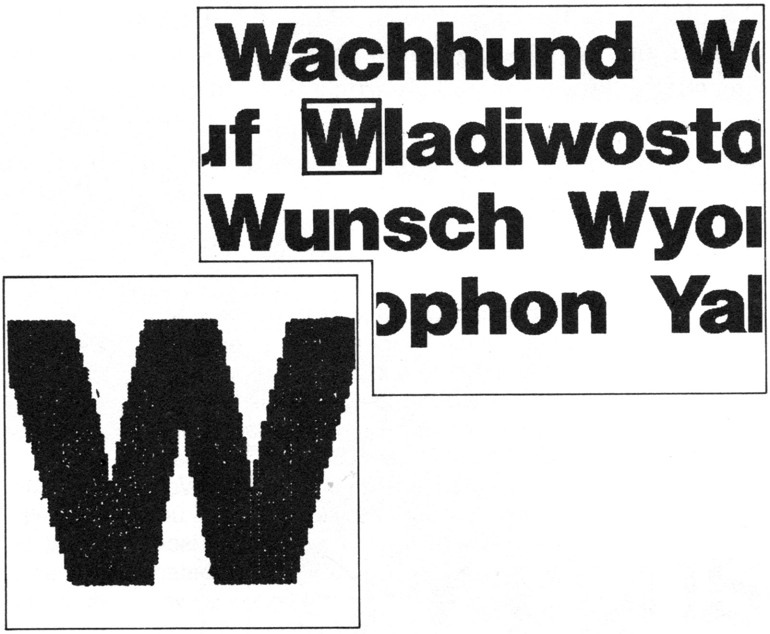


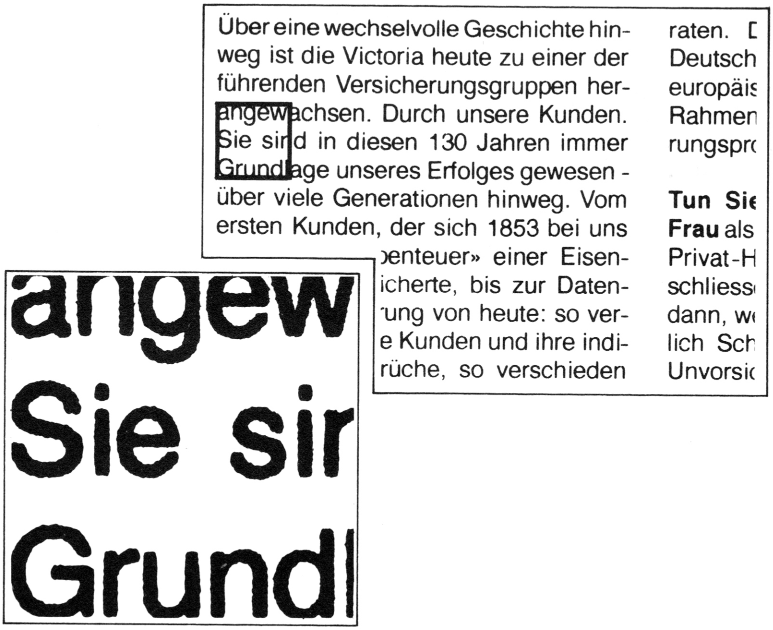
Fig. 26
Electrostatic printing seems inferior to laser quality.

Laser Printers

300 to 600 lpi

Working on a principle similar to electrostatic printers, laser printers use a sharply-focused laser light beam which can be rapidly turned on and off. The beam directed to a sensitive drum element charges that surface electrostatically in order to attract highly charged ink particles to appropriate areas. In general, resolutions range from 12 to 24 lines/mm or 300 to 600 lines/inch in both X and Y direction.

More recent developments provide resolutions up to 1,200 lines/inch (dots per inch) on plain (non-photographic) paper with newly developed ultra-fine grain toner ink powder with the average printer gradually moving upward to 600 dpi (Canon AX or BX engine and others). There is a great deal of speculation on the impact of new 1,000 and 1,200 dpi laser printers on the existing photo imagesetters of that range. Current thinking is that toner particle size and distribution will remain a limiting factor in plain paper printers thus guaranteeing a continuing market for photographic machines into the foreseeable future.



Über eine wechselvolle Geschichte hin-
weg ist die Victoria heute zu einer der
führenden Versicherungsgruppen her-
angewachsen. Durch unsere Kunden.
Sie sind in diesen 130 Jahren immer
Grundlage unseres Erfolges gewesen -
über viele Generationen hinweg. Vom
ersten Kunden, der sich 1853 bei uns
)enteuer» einer Eisen-
icherte, bis zur Daten-
'ung von heute: so ver-
e Kunden und ihre indi-
rüche, so verschieden

raten.
Deutsch
europäis
Rahmen
rungspr

Tun Sie
Frau als
Privat-H
schliess
dann, w
lich Sch
Unversic

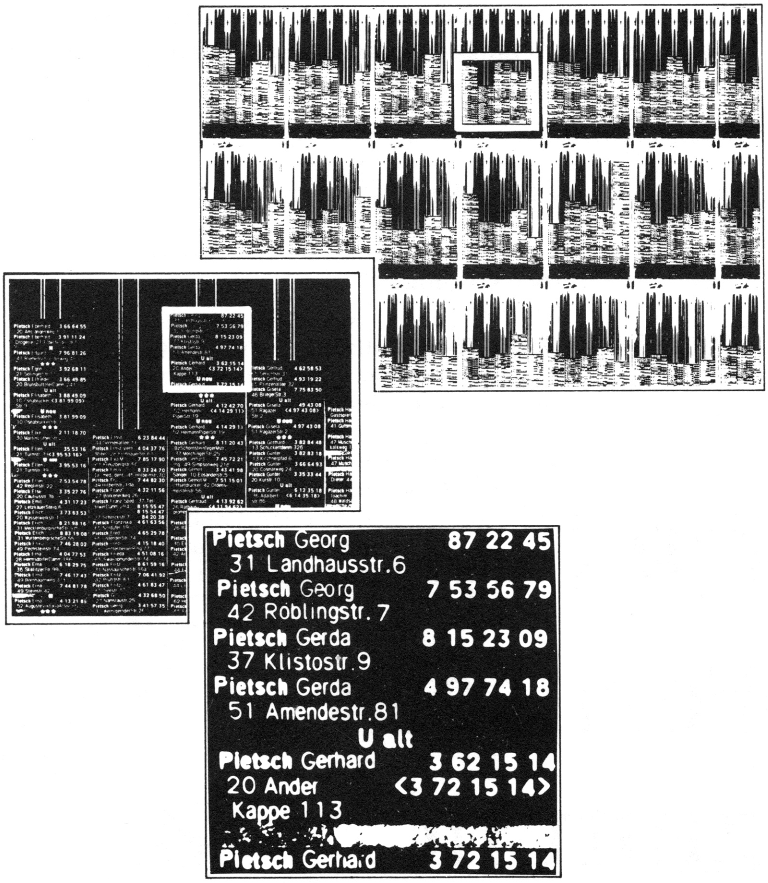
angew
Sie sir
Grundl

Fig. 27
Typical laser printer reproduction of type

Microfilming devices

In microfilming devices control signals are generated as in drawing machines. A very sharply focused light beam writes text relatively slowly in the form of small, straight strokes. Resulting resolution is about 300 lines/mm or 7500 lines/inch after photographic reduction.

7500 lpi



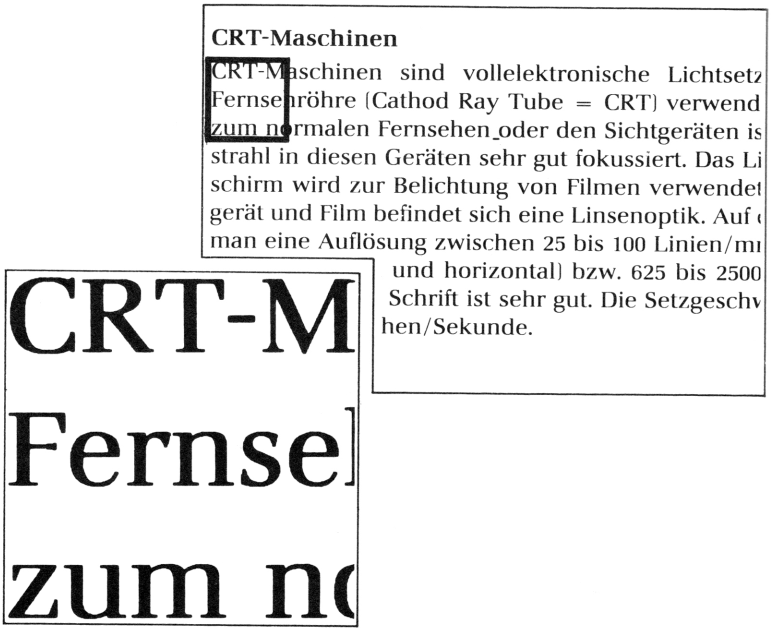
Pietsch Georg 87 22 45
31 Landhausstr.6
Pietsch Georg 7 53 56 79
42 Röblingstr. 7
Pietsch Gerda 8 15 23 09
37 Klistostr.9
Pietsch Gerda 4 97 74 18
51 Amendestr.81
U alt
Pietsch Gerhard 3 62 15 14
20 Ander <3 72 15 14>
Kappe 113
Pietsch Gerhard 3 72 15 14

Fig. 28
Microfilming devices achieve fantastic resolutions. This is a double enlargement!

CRT machines

These machines are fully electronic phototypesetting devices making use of cathode-ray tubes (CRTs). The essential difference from normal TV screens or display terminals is that the electron beam is extremely sharply focused. The light on the tube's display surface is used to expose film. A complex optical lens system is positioned between the display and the film. On film, resolutions are normally 25 to 100 lines/mm, or 625 to 2,500 lines/inch, consequently type reproduction is very good. Typesetting speeds range upward to 2,000 characters/second and more.

1200 lpi



CRT-Maschinen

CRT-Maschinen sind vollelektronische Lichtset
Fernsehröhre (Cathod Ray Tube = CRT) verwend
zum normalen Fernsehen oder den Sichtgeräten is
strahl in diesen Geräten sehr gut fokussiert. Das Li
schirm wird zur Belichtung von Filmen verwende
gerät und Film befindet sich eine Linsenoptik. Auf
man eine Auflösung zwischen 25 bis 100 Linien/m
und horizontal) bzw. 625 bis 2500
Schrift ist sehr gut. Die Setzgesch
hen/Sekunde.

CRT-M
Fernse
zum n

Fig. 29
Example of type reproduction on a CRT machine

Laser film recorders

In laser film recorders or imagesetters, laser light is used to expose film directly. Control signals move one or more light spots across the film switching it appropriately on or off. Resolution and composing speed are comparable with CRT machines, although most are faster. For the first time, these machines can rotate characters arbitrarily. Earlier CRT machines are also able to do this (theoretically), but in actuality, to our knowledge, this has only been realized in one machine as of this writing. The resolution is typically between 50 to 100 lines/mm or 1,250 to 2,500 lines/inch, with newer machines reaching over 3,000 lines/inch.

2400 lpi

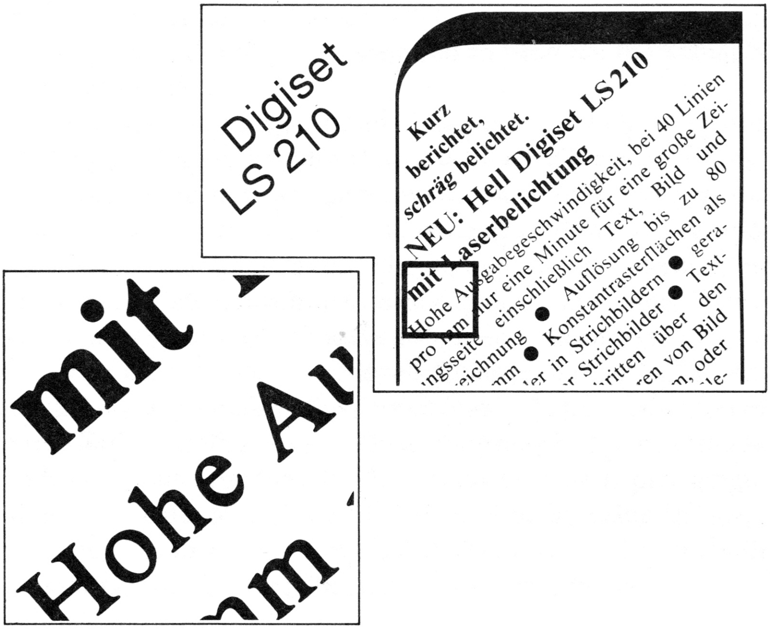

Digiset LS 210
Kurz berichtet, schräg belichtet.
NEU: Hell Digiset LS 210 mit Laserbelichtung
Hohe Ausgabegeschwindigkeit, bei 40 Linien pro mm nur eine Minute für eine große Zeitungsseite einschließlich Text, Bild und
Auflösung bis zu 80
Konstantrasterflächen als
Strichbildern ● gera-
Strichbilder ● Text-
über den
von Bild
oder
mit
Hohe Au
mm

Fig. 30
The Digiset LS 210 is the best laser film recorder ever made! But, maybe, too expensive.

Other typesetting machines

Typesetting machines are available using other methods of operation, e.g., ink jet printers and laser vaporization machines to directly expose printing plates. While not suggesting any lesser importance of such devices, we limit our discussion to a brief summary of the ways and means of displaying digitally stored letters.

When is outline coding necessary?

All machines require their own particular type of digital coding. Correspondingly, the differing resolutions favor specific coding techniques: for lower resolution display terminals and dot matrix printers, bitmaps are used. For output resolutions around 20 lines/mm, scanline coding is common, and for very high resolutions, character outline coding. The various coding schemes stem from a necessity to keep type storage requirements within a machine as economical as possible in all cases.

3.3 NC machines

The term «NC machine» (numerically controlled machine) was chosen as the group name for the machines listed here. These machines share this feature: they move a «pen» in the X and the Y direction simultaneously. Differentiation is made in the actual writing instrument:

An impressive variety of NC machines

Pen	Drawing machine or plotter
Knife	Foil cutting machine
Router	Routing machine
Engraver	Engraving machine
Flame	Flame cutting machine
Laser beam	Laser cutting- or laser engraving machine
Waterjet	Beam cutting machine (pressure cutting machine)
Sandjet	Stone engraving machine

Control signals are treated similarly to that of direct VDTs and microfilming machines. In NC machines however, relatively powerful motors are required to produce the actual X and Y movement.

A benefit of using the same sort of control signal means that digital storage of type is the same for all of these machines.

Plotters, in fact, are also capable of writing typographic characters, but are extremely slow compared to laser composers. «Typesetting speed» of 1 character per second is common in plotters. Newer machines boast capability of writing instruments moving at speeds between 1 cm/sec and 1 m/sec.

The early 1950's saw development of numerically controlled machine tools in the United States. The introduction of large flatbed plotters from 1950 to 1960 may be traced initially to Konrad Zuse and the ARISTO company. Zuse kept to a rigid digital development of X and Y axis drives by using DC stepping motors. In 1959 the first «ARISTO Coordinatograph with numerical continuous path control» was introduced.

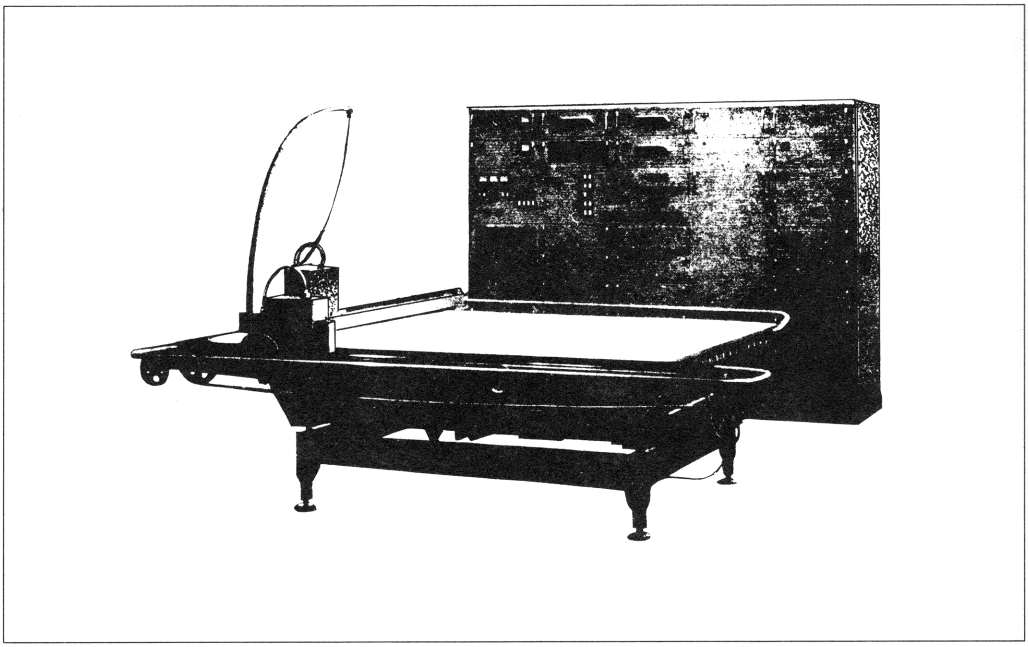


Fig. 31
Aristomat with continuous path control, 1959

Flatbed drawing machines of today are evolutions of these earlier machines and have working areas of about 1 m^2 up to 2 m^2.

These machines are highly accurate, to 1/100 mm and are expensive for this reason. Smaller plotters, while less expensive, have working areas corresponding to A3 format, and a drawing accuracy of 1/10 mm.

1/100 mm means 2540 lpi!

Three representative NC machines are described on the following pages.

Plotters

Ink pens, available in multi-colors, are the writing mechanism in these machines. Smaller plotters have a resolution of approximately 0.1 mm up to 0.025 mm or 0.004 inch up to 0.001 inch, corresponding to 10 to 40 lines/mm or 250 to 1000 lines/ inch in larger machines.

250 lpi

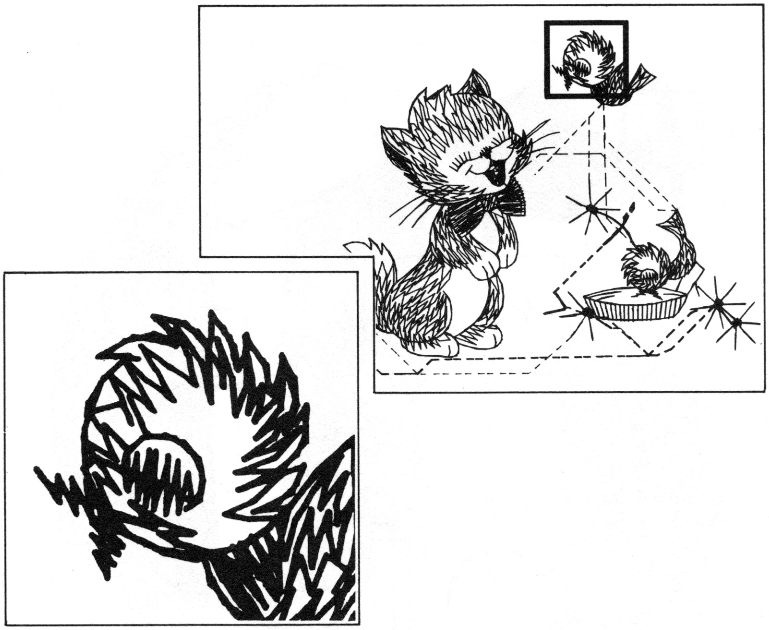


Fig. 32
Plotters are very good for checking graphic data.

Flatbed drawing machines

5000 lpi

The larger flatbed plotters actually have earned the description «drawing machine». They have resolutions of 0.025 mm up to 0.005 mm or 0.001 inch up to 0.0002 inch, corresponding with 40 to 200 lines/mm or 1000 to 5000 lines/per inch. From these specifications it is clear that these are more than mere plotters!

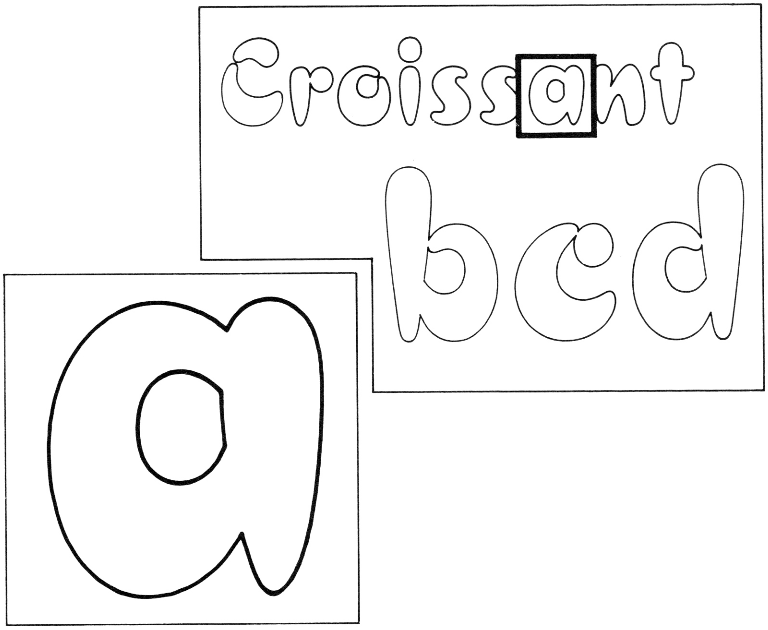


Fig. 33
Drawing machines deliver end products - finished drawings which are ready for filing.

Foil or vinyl cutting machines

Most flatbed drawing machines, with addition of a rotating cutting tool, can also be used to cut laminated films. Here we include a very clean, large cut letter to illustrate the precision which cutting can achieve.

These machines are used with the SIGNUS system for sign production.

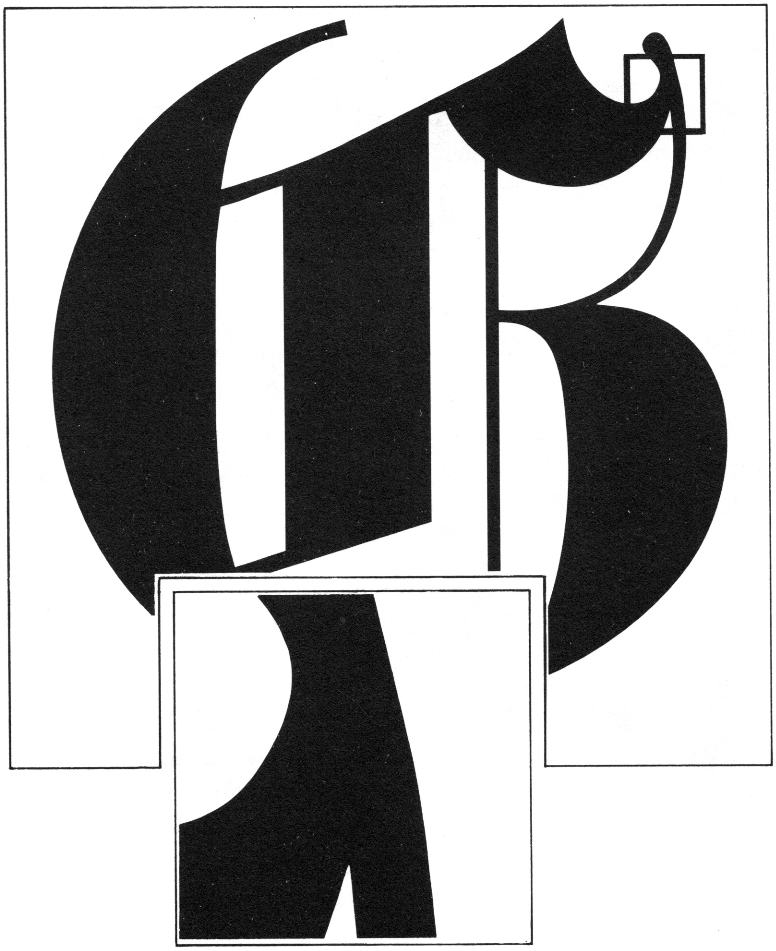
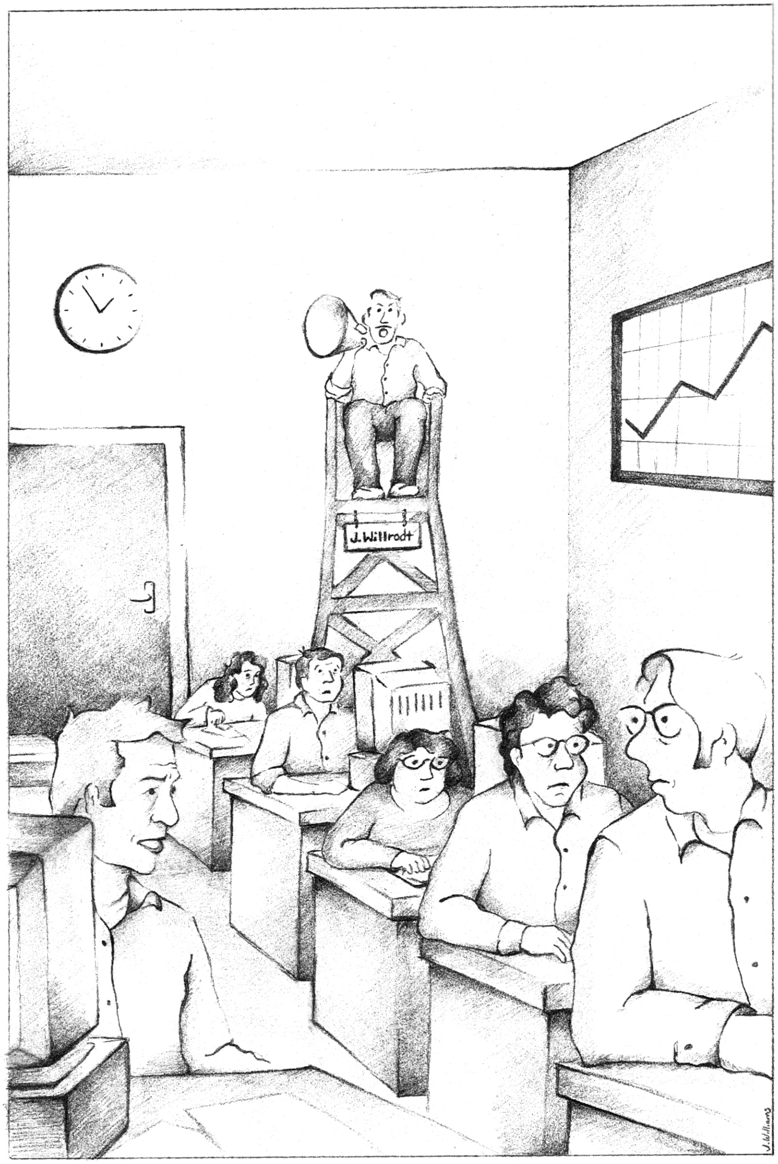


Fig. 34
Foil cutting machines have a resolution of 2540 lpi – and that with characters of up to two meters in size!


J. Willrodt
J.Williams

«Somehow,
I pictured
TypeWorks 2
production
differently!»

Requirements of typesetting

Different machines and associated electronic modules within the machines each require different coding. Universally, digital storage for any machine must be accomplished so type coding consumes as little memory space as possible, and also can be read and processed as quickly and directly as possible. This necessity is best achieved by optimum matching of type code to electronics used. The requirements imposed by the machine on a digital format of typefaces pertain, for the most part, to the code. The actual display of texts, the actual typesetting process, increases the requirements however.

4.1 Type structure

The initial requisite is that data stored, respective to each individual character, be accessible. To this end all characters are assigned discrete addresses all listed in the character address table.

Secondly, the left side bearings, right side bearings, and total widths of the characters must be accessible independent of image data. Widths are therefore stored as separate data tables known as «width tables». Accents present slightly more of a problem when treated as «floating accents», related to characters as marked positions, arrived at only during actual typesetting. This is accomplished employing either a typesetting rule whereby an e and ´ produce an é, or, with the aid of markings in e and `, typesetting produces an è. Fixed accented characters using separate letters for: é, è, ë, ñ, á, ä, ü, etc., are preferable as accented letters are unique, and could, as such, be checked separately by type designers at a type design stage.

Floating or fixed accents

Ligatures are also important – as mentioned in Chapter 2. They were devised to produce improved appearance of commonly used letter combinations, and often greatly enhance the overall appearance of well spaced text.

Ligatures and swash characters

Additionally, there exist swash characters, such as an «e» with an extended embellished stroke on its terminal, placed at the end of a word rather than a normal «e», used at the beginning and middle of words.

These three variants – accents, ligatures and swash characters – have traditionally created problems when entering text (on English keyboards for VDTs) where only normal individual characters are accessed by specific keys. Application software is then used to find the new addresses for combined characters as fixed accent letters, ligatures or swash characters. Not all text systems are capable of accomplishing this at the moment.

Understanding this, it should be clear why a comprehensive description of a typeface (see also Appendix B) is required. It may be necessary to store particular additional parameters:

- for superior indices (superscripts) and
- for lower indices (subscripts).

Furthermore, «kerning» and «touching» tables also accompany a particular typeface. Kerning was also discussed briefly in Chapter 2.

Some machine producers and type manufacturers have yet to deal adequately with kerning. Some, though not all, companies produce so-called «aesthetic boxes», to provide correction values for certain important letter pairs. For example, the illustration below lists letter pairs for which the Berthold Company provides kerning values. As a rule, these are negative values used to correct (reduce) traditional letter spacing by use of setwidths.

AT AY AV AW Ay Av Aw A’ FA F. F, TO TA Ta Te
To Ti Tr Tu Ty Tw Ts Tc T. T, T: T; T- LT LY
LV LW Ly L’ PA P. P, VA Va Ve Vo Vi Vr Vu Vy
V. V, V: V; V- RT RV RW RY f’ WA Wa We Wo Wi
Wr Wu Wy W. W, W- YA Ya Ye Yo Yi Yp Yq Yu Yv
Y. Y, Y: Y; Y- r’ r. r, r- y. y, v. v, w. w,
’s

At URW a completely different concept is used: the distance between any two letters is a discrete value, i.e. the kerning table for 100 characters has a length of 100 × 100 = 10,000 words; for 250 characters its length is 250 × 250 = 62,500 words. This means that aesthetic tables are considerably smaller than the «long kerning tables» provided by URW where such specific tables are prepared for every type font.

Long kerning tables

Additionally, at URW we understand «kerning» as to be not negative only, but also positive. Positive kerning gives a little repulsion to pairs «v w» or «v y»

In addition to kerning values, there are also correspondingly structured touching tables allowing spacing between letters to reduce spacing still further, letting them touch. Safety zones ensure that, for example with an HIH set in a sans serif typeface, the letters do not touch.

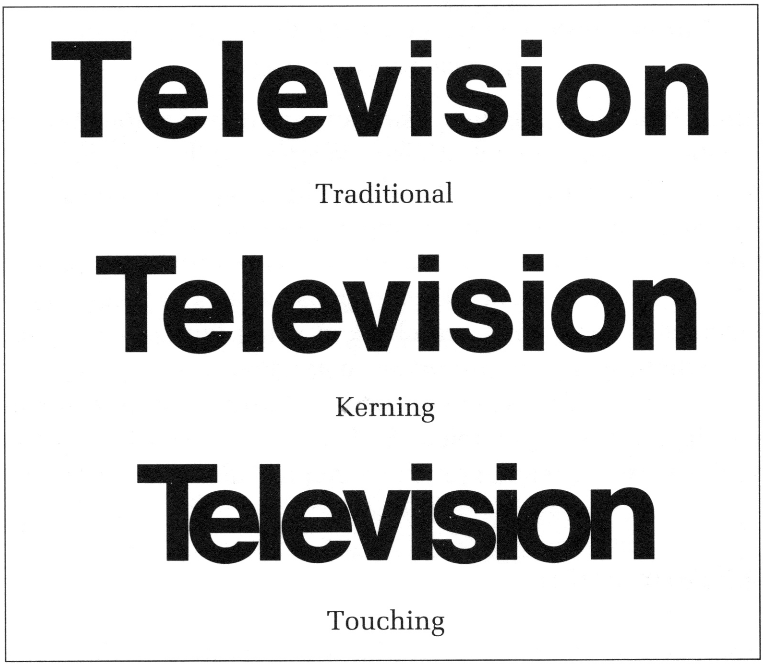


Fig. 35 Traditional, kerned and touching type

The overlapping table allows pairs of letters to approach each other so that they overlap – known also as «sexy» setting. Blending is where black areas of the letters blend into one another. Where thin white edgelines outline the covering letter is called «overlapping».

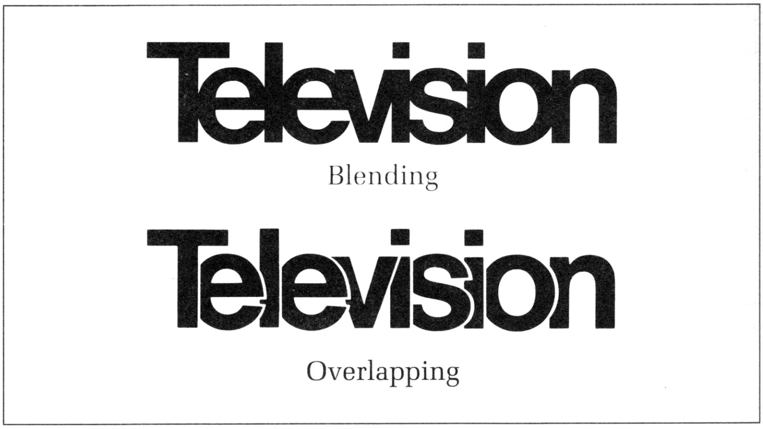


Fig. 36
Type can even blend or overlap.

In normal use, touching values are applied only when setting headline type (cap height above 1 cm at normal reading distance) for optically correcting kerning values. One uses the rule:

«The larger the letters, the greater the negative kerning, approaching touching».

Since 1983 URW has been using an algorithmic automatic calculation system, nine years in development, dealing not only with traditional spacing but also with kerning, touching and overlapping tables; a very complex system. In 1992, it ended with the invention of »Kerning on the Fly», providing a speed of calculation of about 100 pairs per second. The results of its calculated application are checked using typeset examples, and where necessary, are refined manually.

In Appendix R we have described the data structure for all spacing, and in Appendices S, T, U and V the set-up of the width, kerning, touching and overlapping table.

4.2 Form setting

Form setting refers to setting lines not running horizontally straight, i.e., sloping, circular, wavy or etc., although the same term is also applied to mathematical formula tables, table typesetting, typesetting of graphic elements (lines, frames, vignettes and generally all possible configurations used in successful setting of advertisements). The setting of type along contours of geometric elements, or otherwise in combination with such elements is worthy of our attention.

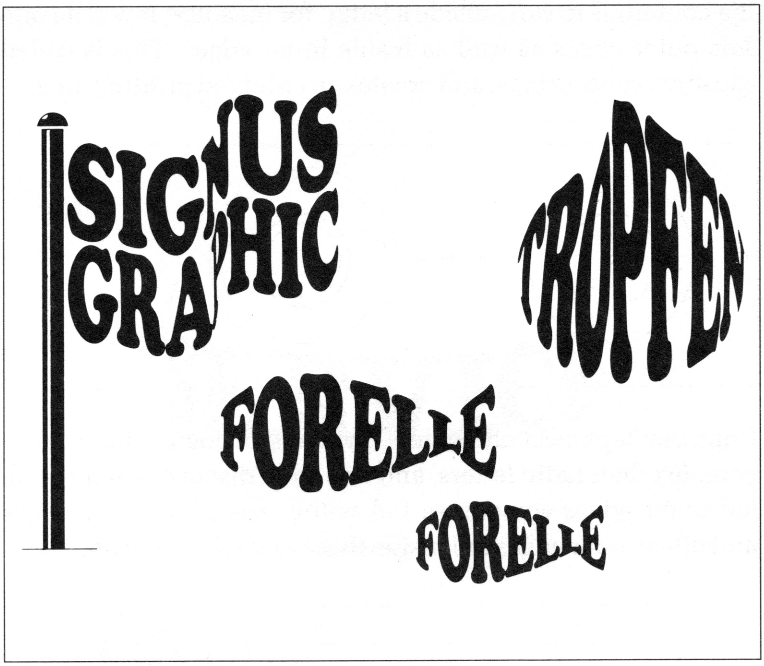

SIGNUS
GRAPHIC
TROPFEN
FORELLE
FORELLE

Fig. 37
Type set along lines, frames, forms

Form setting requires this additional data:

- Body size in original database
- Cap height
- x-height
- Stroke thickness of type (weight)
- Optimum size range for the typesetting
- Width (traditional, kerning, touching, overlapping).

In addition, all codes must identify which inner edge relates to which outer edge, especially when contouring.

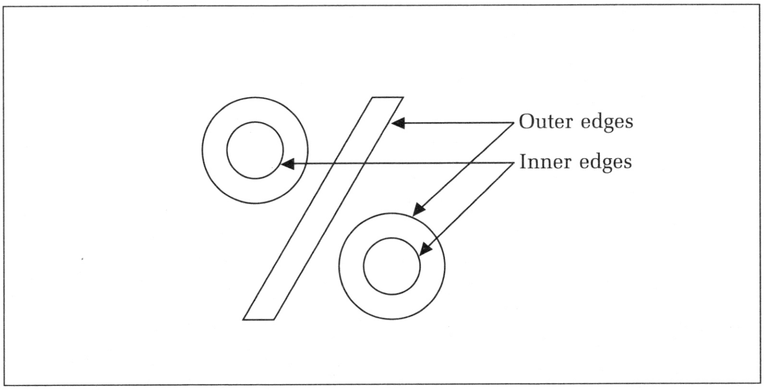
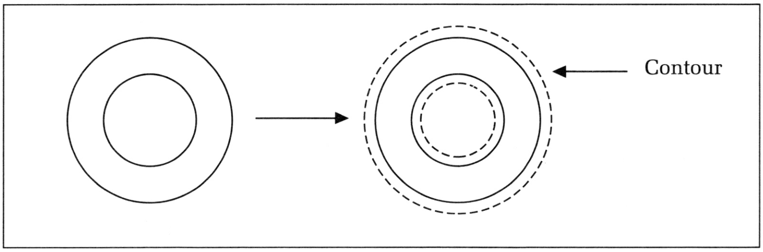

Outer edges
Inner edges

Fig. 38
The contours of the characters have a hierarchy.

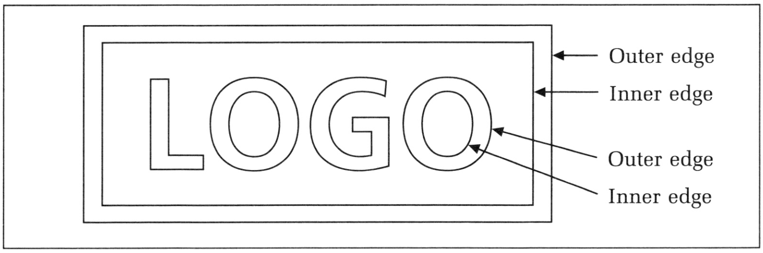
If a contour is to run outside a letter, for instance, it will lie outside outer edges as well as inside inner edges. This is called «positive contouring», and creates an enlarged printing area.

Fig. 39
Contouring produces new inner and outer contours.



Consider Japanese characters, symbols or logos, those more complex than Latin letters, and one soon discovers that so-called outer edges sometimes fall within so-called inner edges and other inner edges fall inside these as well, and so on.

Fig. 40
In logotypes, many hierarchical contours are produced.



Setting to shape thus presents an even greater demand on the digital format. It must be distortable, adaptable to specified, yet in principle, arbitrarily chosen external shapes. In setting on a circle, for example, normally horizontal lines are bent to the radius of the setting circle.

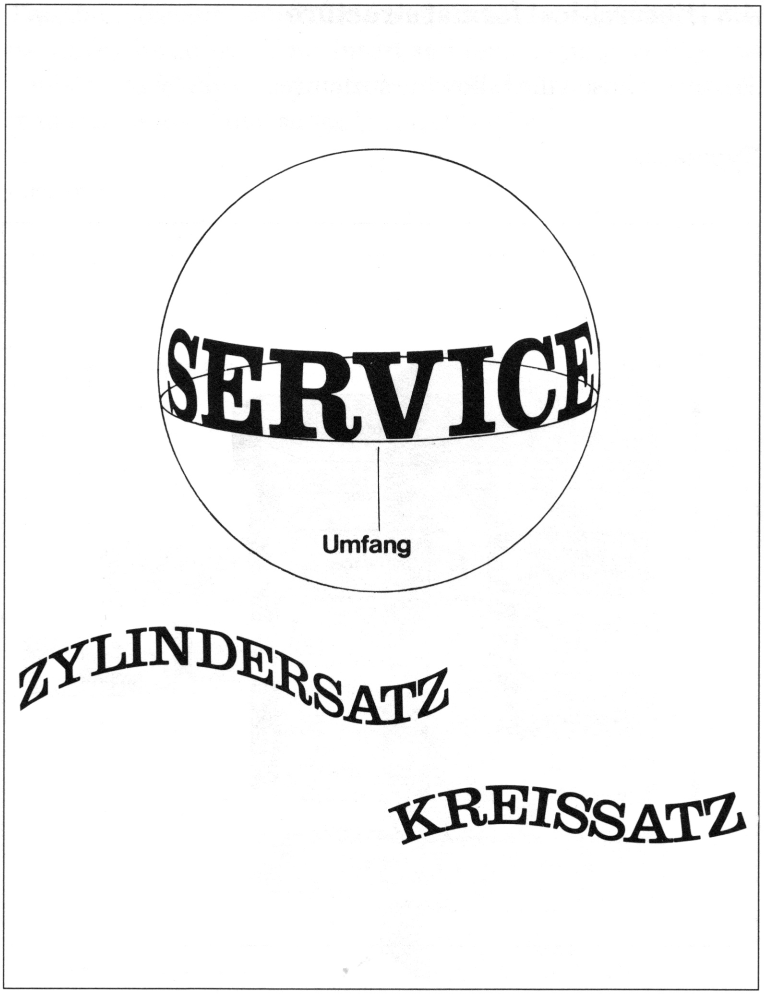

SERVICE
Umfang
ZYLINDERSATZ
KREISSATZ

Fig. 41
Characters are bent and adapted to the required shape.

4.3 Hierarchical format structure

We have chosen the following structure:

Typefaces

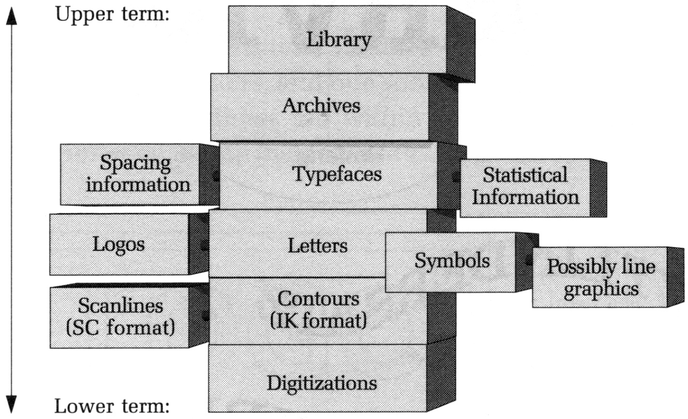

Upper term:
Library
Archives
Spacing information
Typefaces
Statistical Information
Logos
Letters
Symbols
Possibly line graphics
Scanlines (SC format)
Contours (IK format)
Digitizations
Lower term:

Fig. 42
Typefaces cannot be stored without a hierarchy.

This data structure simplifies manipulation of related data. For example, the copy, delete, insert and replace functions can be applied especially to a specific hierarchical level, being limited to the elements of that single hierarchical level.

Graphics

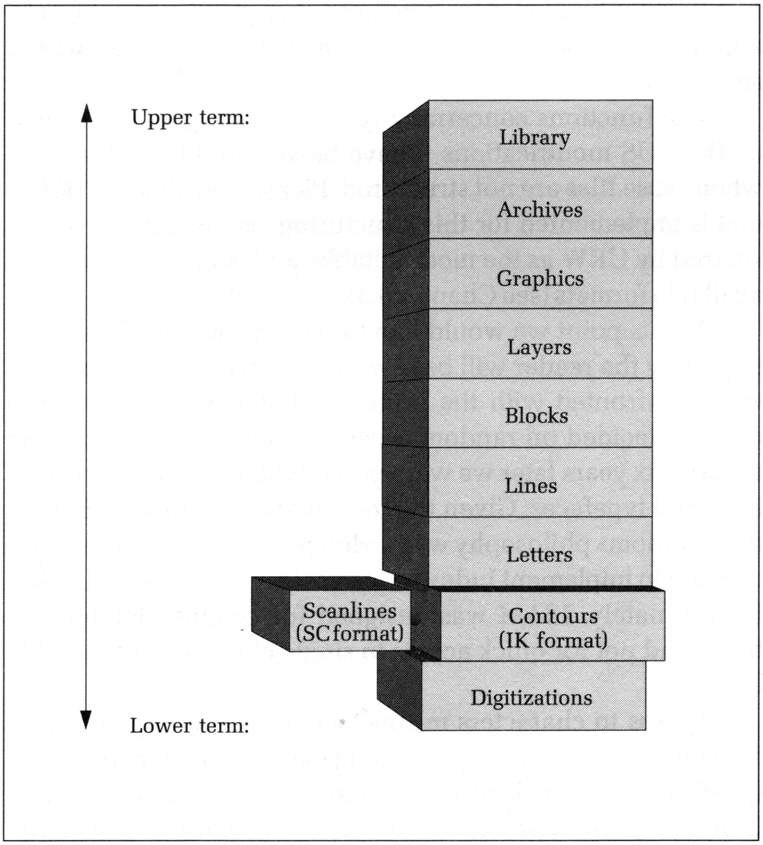

Upper term:
Library
Archives
Graphics
Layers
Blocks
Lines
Letters
Scanlines (SC format)
Contours (IK format)
Digitizations
Lower term:

Fig. 43
The hierarchy in graphics and texts makes data manipulation easier.

In Appendix G is an exact description of the basic typeface structure. In Appendix H is the structure of a typeface in the IKARUS format. Within the framework of typeface storage, a special structure is required for the typeface level and below. The actual host file management system should deal with levels above typeface.

The graphics storage requires a far more complex structure than that required for typefaces, though, in other respects, a graphic is treated hierarchically as is a typeface or, with a simple graphic, as a character.

The functions concerning typefaces and graphics – such as IKARUS modifications – have been found to be too slow when those files are not structured. Please note that the IK format is implemented for this structuring, as this format is considered by URW as the most suitable, and is the shortest of all available formats (see Chapter 6.3).

Random access to typefaces on disks?

At this point we would like to indulge in a little history – hopefully the reader will bear with us. Nineteen years ago we were confronted with the problem of storing typefaces. We quickly decided on random access because of the quick disk access! Six years later we were posed with the problem of storing Kanji typefaces. Given the then available 5 to 10 MB disks our «random» philosophy was no longer suitable. We therefore decided to implement **i**ndex **s**equential **a**ccess **m**ethod (ISAM). Unfortunately, ISAM was designed for commercial applications, and not for quick access to single characters of variable length.

Access to characters means, above all, loading one typeface into RAM and then processing many characters from this typeface, and not loading numerous individual characters from different typefaces from the disk. Normal ISAM file management systems are not fully capable of managing this relationship between characters and typefaces but are designed for searching for individual elements. Our element is – and this we learned the hard way – the typeface, the large graphic, the collections of frames for images or line elements, and not every individual figure such as the character of a typeface, an element of a graphic, or a specific frame or line.

The developments in semiconductor memories and the advent of 32 bit microcomputers ideally suit our current, revised storage concept. At last we are able to store our critical mass, namely 20 to 100 KB for a Latin typeface as one unit, which was impossible with the previous 64 KB address barrier: a typeface is loaded as soon as only one character from this typeface is required. We then enclose the IK or IG structures with the typeface or graphic shells, enabling individual access to the desired hierarchical levels such as point, contour, etc.

Memory capacity is no longer a problem.

We recommend this procedure to all those who work with typefaces and graphics.

Computer, like you and me

Translation

An American professor gave a presentation on his most recent findings in Tokyo. He is well-renowned in his special scientific field of research. He got an interpreter at his side and started his intervention. It took him about 20 minutes to set up the first important statements. Then he stopped to give the Japanese interpreter to translate. He could do it with two sentences, astoundingly. Then the professor continued with the second brouse of phrases from the front edge of technology. Again he made a pause after 20 minutes for translation. This time, the Japanese could do it with one sentence only. Finally, for another 20 minutes, the professor announced more teaching, climbed up the summit of his research and summarized at the end. Also, this time, the Japanese interpreter had to do his job and - needed only two words for translation. After that, the audience seemed to be very, very pleased, applauded passionately, and the conference went on.

The professor was pleased and satiesfied, stayed for the day, listened to others, and finally went to a special invited dinner for speakers.

As the three parts of translations were that short, he wondered about the intrinsic power of the Japanese language. How overwhelming, what a new possible field of research.

So, to learn more about it, having dinner and sitting side by side with another Japanese professor, he asked him about the pleasant interpreter and what he translated to the Japanese. He just wanted to know a little bit about the secrets of the Japanese language.

«Oh», said the Japanese professor, «excuse me, but he said at first: ‹The American is a nice man, just that he talks too loud and too much.› Second: ‹We knew all about he is talking.› And third: ‹Forget it.› »

Formats

As capabilities of electronics increased in the past so did the comprehensiveness of type data digital information. On one hand, maintaining compact data required more highly-developed control electronics, while, on the other hand, the expanded requirements of typesetting demanded even more extensive electronics plus higher memory capacities to contain type information.

Earlier, data was stored as easy-to-deal-with bitmaps or run lengths, and later as outline codes (vectors), followed by still more complex formats.

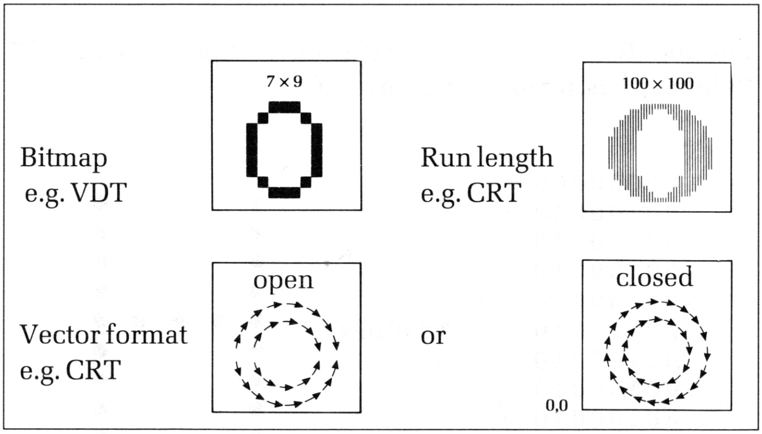


Fig. 44
First bitmaps are stored, then run lengths, later vector lines.

A substantial number of machines are currently available using digital typefaces with a correspondingly large number of digital typeface formats. This variety necessitates categorization according to these characteristics:

• method	bitmap bytemap (halftones) run length vectors

curves
segmentation
Metafont
- coordinates absolute
incremental
- code simple
complex

These last two characteristics – coordinates, code – do not necessarily denote a difference of format. Absolute coordinates can be converted to incremental and vice versa. Similarly, complex coding can be calculated back to its simple original form (and, likewise, vice versa).

In Chapter 5.3 this is discussed in greater detail. First, let us consider methods to establish a format review.

5.1 Review of formats

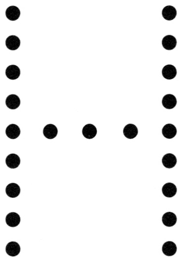
5.1.1 Bitmaps (bytemaps)

Consider the letter «H» for display on an alphanumeric VDT. Its bitmap has, for example, the form:

```
00000000
00100010
00100010
00100010
00100010
00111110
00100010
00100010
00100010
00100010
```

8 × 10 bits

10 bytes/char.



It would therefore seem obvious to store its 80 bits in 10 bytes by scanning the bitmap from left to right horizontally, line by line from top to bottom. In so doing, one may set zeros equal to black, ones equal to white or vice versa (reverse video).

The format of a typeface to be typeset, monospaced, on an alphanumeric VDT (first generation) is thus extremely simple:

Every 10 bytes a new character begins. In a device displaying 128 characters, a 1280 byte memory is required. Addresses for image information are basically «ASCII address times 10».

Gray characters

With newer second generation display terminals and TV captions bytemaps rather than bitmaps are needed. Depending upon the number of halftone pixels at the boundary between black and white, one may have quarterbytes (4 steps of gray), halfbytes (16 steps of gray) and bytes (256 steps of gray). To transform a bitmap into a halfbytemap, 4 × 4 bits are gathered, counted out and stored. There is, however, a limitation: 15 and 16 bits both produce a halfbyte of the highest value 15 (see also the following diagrams). The transformation of halfbytes back into their original bits is not possible (see lit. Warnock).

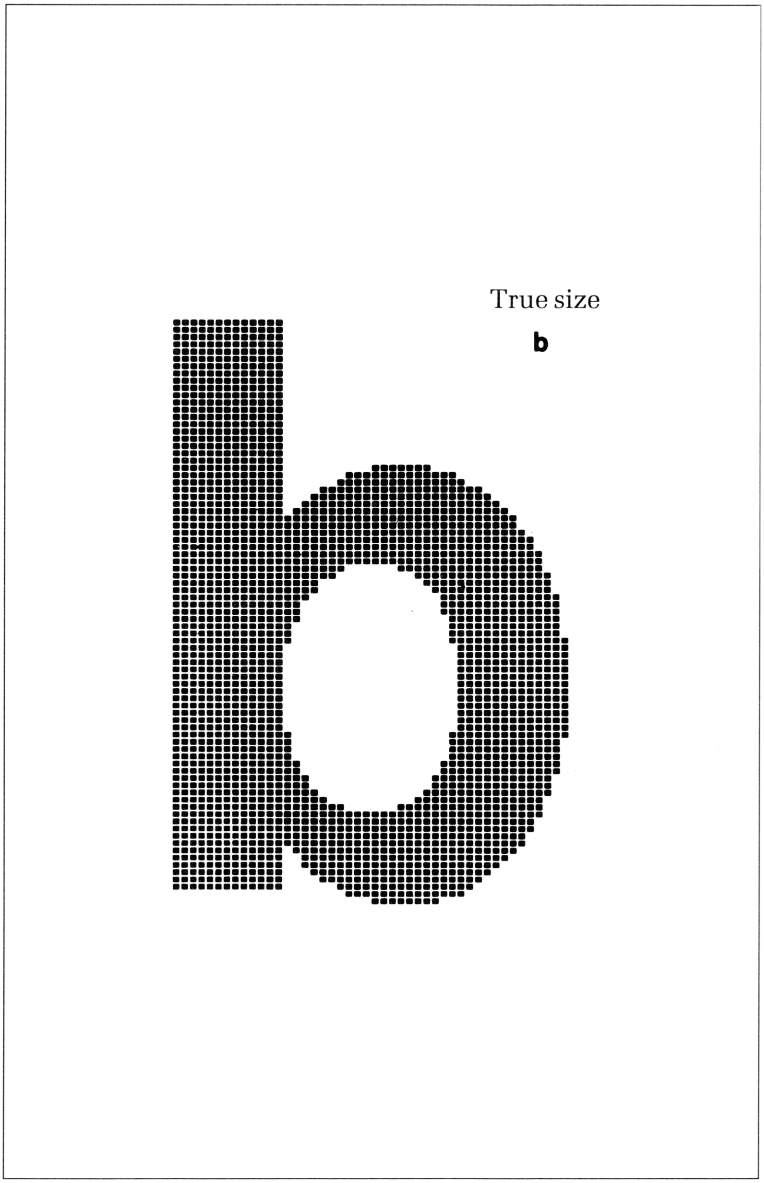


Fig. 45
Bitmap of lower case b at a raster of 100× 120 for the em.

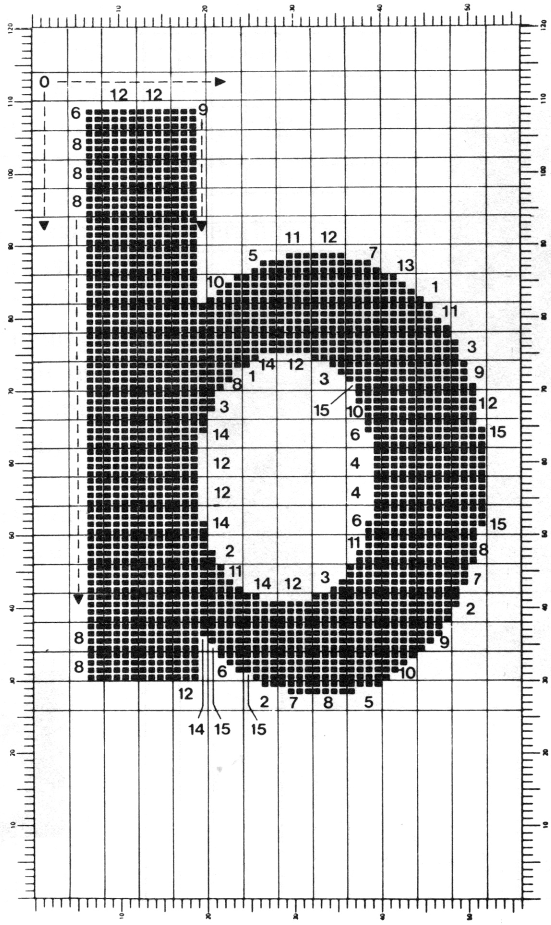

10
20
30
40
50
120
110
100
90
80
70
60
50
40
30
20
10
0
12 12
9
6
8
8
8
11 12
5
7
13
10
1
11
3
14 12
8
1
3
9
12
3
15
10
15
14
6
12
4
12
4
14
6
15
8
2
11
7
14 12 3
2
9
8
8
6
10
12
2 7 8 5
14 15 15

Fig. 46
Preparation and calculation of a bytemap at a raster of 25 × 30 for the em.

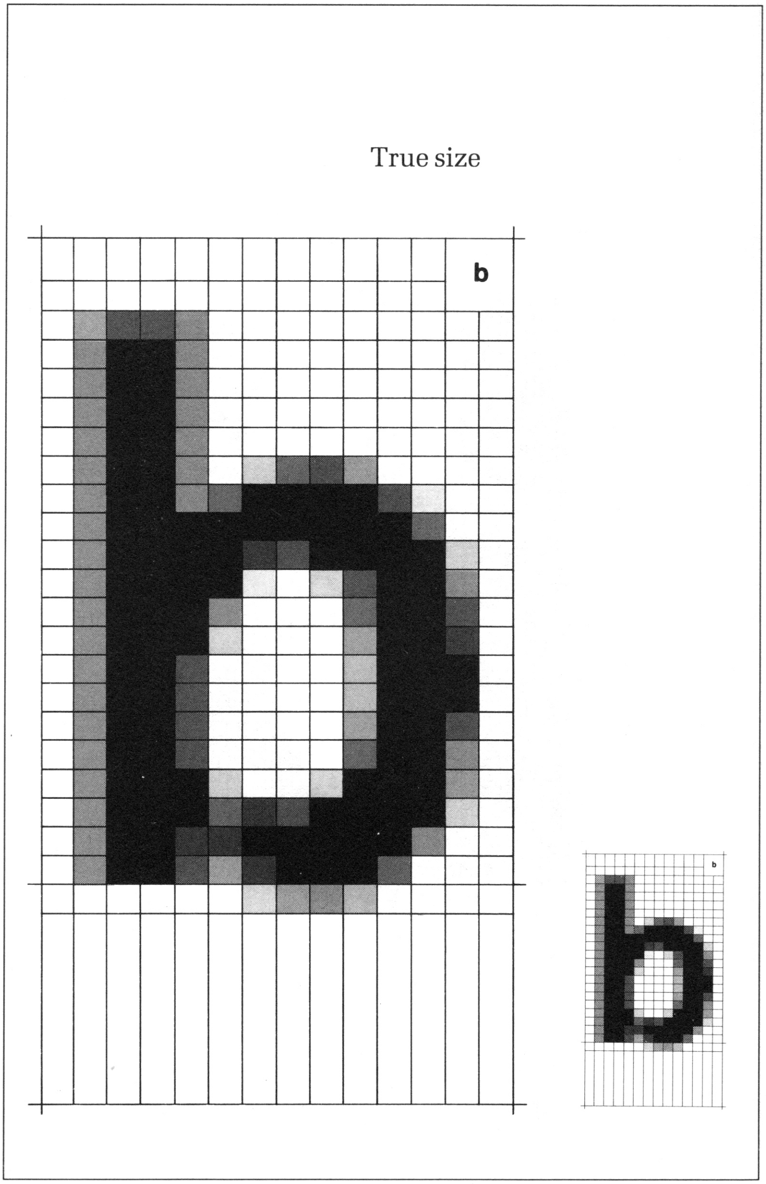

True size
b
b

Fig. 47
Bytemap of a lower case b at a raster of 25 × 30 for the em.

5.1.2 Run lengths

Bitmaps and run length codes are basically similar. As bitmaps get larger and larger a question arises whether one should consider the successive zeros and ones in a line, that is, count them and code their number as run lengths. When displaying letters either horizontally or vertically (columns or rows) in a raster, there occur black and white alternations. Run length coding assumes a line begins external to the letter and the first segment of the line thus always begins white. Black and white then alternate at transitions, not requiring further color coding in black or white.

On average, Latin letters have 3.5 transitions per scanline passing the em and fill approximately 47% of the em width. Thus, for a 32 × 32 bit raster one needs approximately 32 × 0.43 × 3.5 = 48 bytes per letter plus 14 + 6 = 20 bytes for organization and handling of scanlines (administration data), a total of 68 bytes. The bitmap technique requires 14 × 4 = 56 bytes (14 = average character width of 32 × 32) not including the data required for administration. It becomes apparent from the above resolution level upward, run lengths are a shorter coding form than bitmaps.

On average, the characters are half as wide as the em.

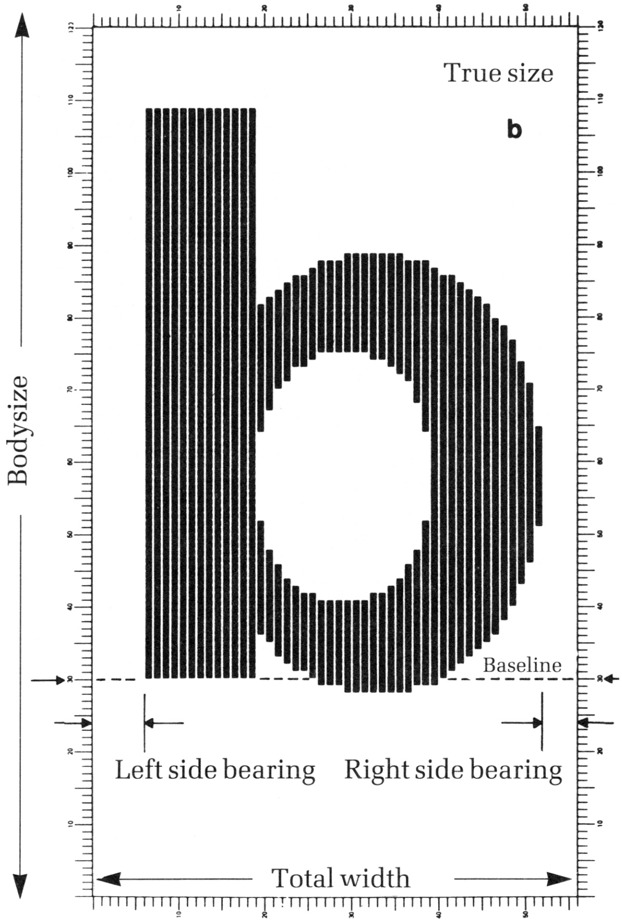

True size
b
Bodysize
Baseline
Left side bearing
Right side bearing
Total width

Fig. 48
Run length of lower case b at a raster of 100 × 120 for the em.

5.1.3 Open vectors

Transitions in neighboring lines have a linear relationship in the first approximation. First, let us consider the «H» in Helvetica type:

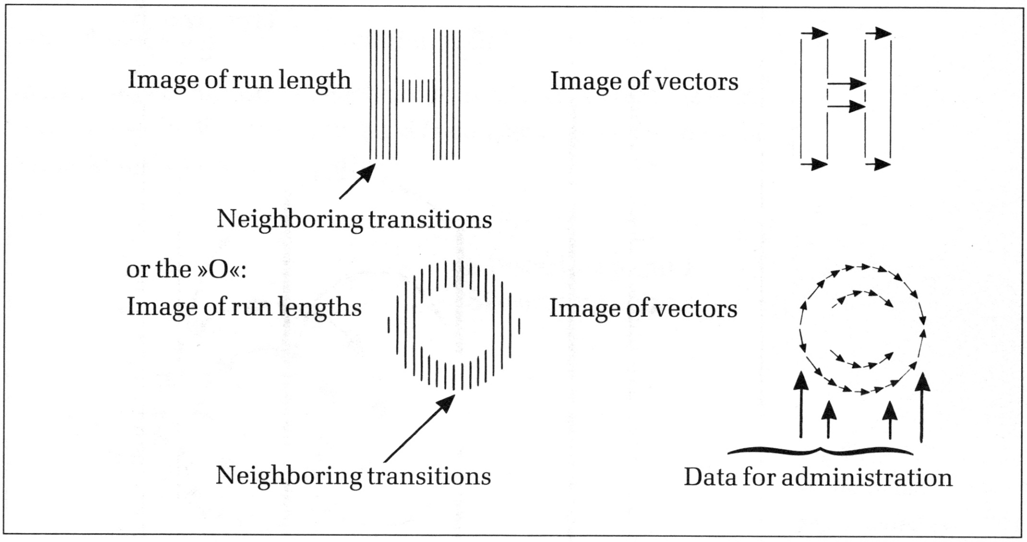


Image of run length
Image of vectors
Neighboring transitions
or the »O«:
Image of run lengths
Image of vectors
Neighboring transitions
Data for administration

Fig. 49
Transition from run length to vector representation

Since vector images consist of fewer straight lines than corresponding run lengths, vector images require less data.

In order to describe a vector, in principle the two coordinates of the beginning point and the two for the ending point are required. Frequently, the ending point of the preceding vector is the same point as the beginning point of the following one. When this is true, one needs only the ending point. As mentioned, description of vectors entails more data in «administration» than used in run length coding, when considering description start and finish data for line pairs as well as the additional electronics required to generate control signals.

Vectors save on storage.

Vectors are also used in generating letters for direct VDTs (first generation graphic VDTs), microfilming devices and line plotters. Where earlier vectors represented edges of black surfaces, now they represent the center of basic strokes.

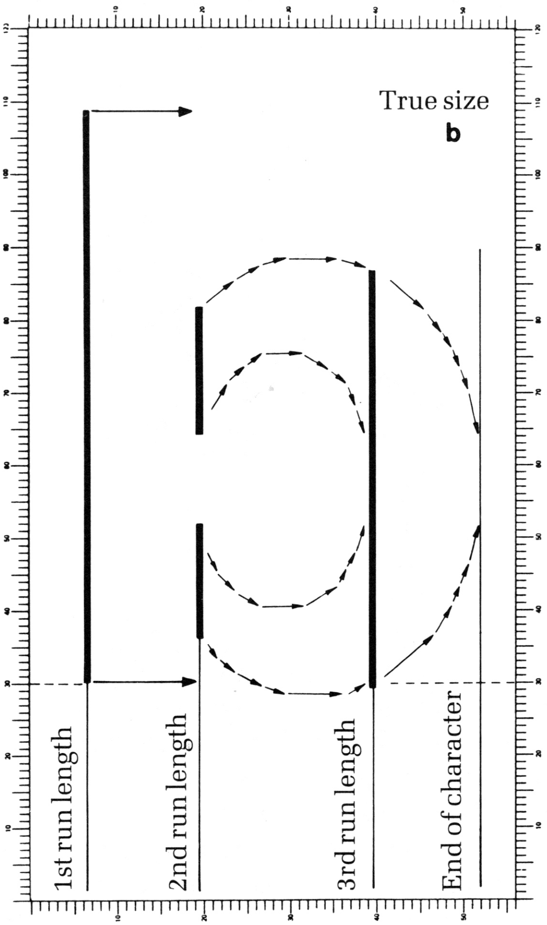

True size
b
1st run length
2nd run length
3rd run length
End of character

Fig. 50
Open vector format of lower case b at a raster of 100 × 120 for the em.

5.1.4 Closed vectors

Once data has been stored in an open vector format, a good deal of electronic effort is required to make them rotatable. To date, no manufacturer has actually managed rotation using the open vector format.

With closed vector formats rotation is easier to achieve:

In a closed vector format, each contour of vectors returns to its point of origin. For example, an «O» consists of two closed-line paths, known as polygon paths:

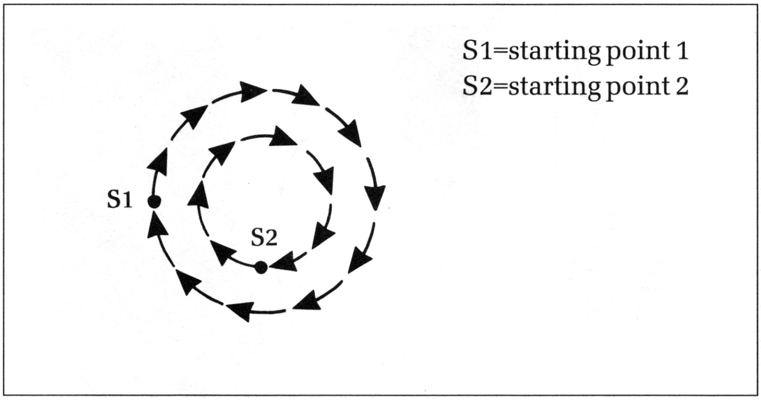


Fig. 51
A character represented by closed vector lines.

Dealing with closed vectors the starting point position for a line is unimportant; meaning that letters stored and processed as closed vectors can be easily rotated or deformed electronically since such transformations only have to remain the characteristic «closed»; there are no «openings» which might be introduced or displaced during the transformation.

In 1982 URW carried out simulating calculations and established that memory capacity requirement is lowest for typefaces coded with 4 bits for X component, 4 bits for Y component of vectors. Memory requirements of both 3 bit and 5 bit coding are much higher. We consider this to be an astonishing and lucky coincidence, as 8 bit (2 × 4 bits for one vector) is an important number in computers. URW uses a special code for the X, Y direction changes and for very long straight lines.

The optimum coding for vector formats is 4-bit.

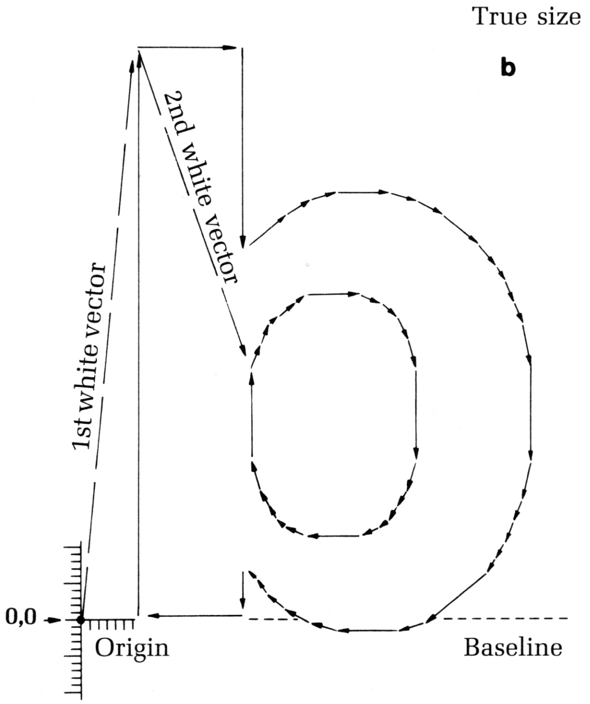

True size
b
2nd white vector
1st white vector
0,0
Origin
Baseline

Fig. 52
VE format of lower case b at a raster of 100 × 120 for the em (each coarser raster is calculable).

Perhaps the characteristics of 4 bits are not so astounding as using 4 bits one is able, without too much effort, to code vector lengths between 1 and 13 (14 and 15 are reserved for special coding).

Curves account for 90% of the character data.

In considering bitmaps for rasters between 100 × 100 and 400 × 400 per em, irrespective of the fineness of the raster at curves, scarcely any curve vector components occur longer than 13.

Disregarding long straight lines, given a relatively long special coding, curves represent more than 90% of the total code required!

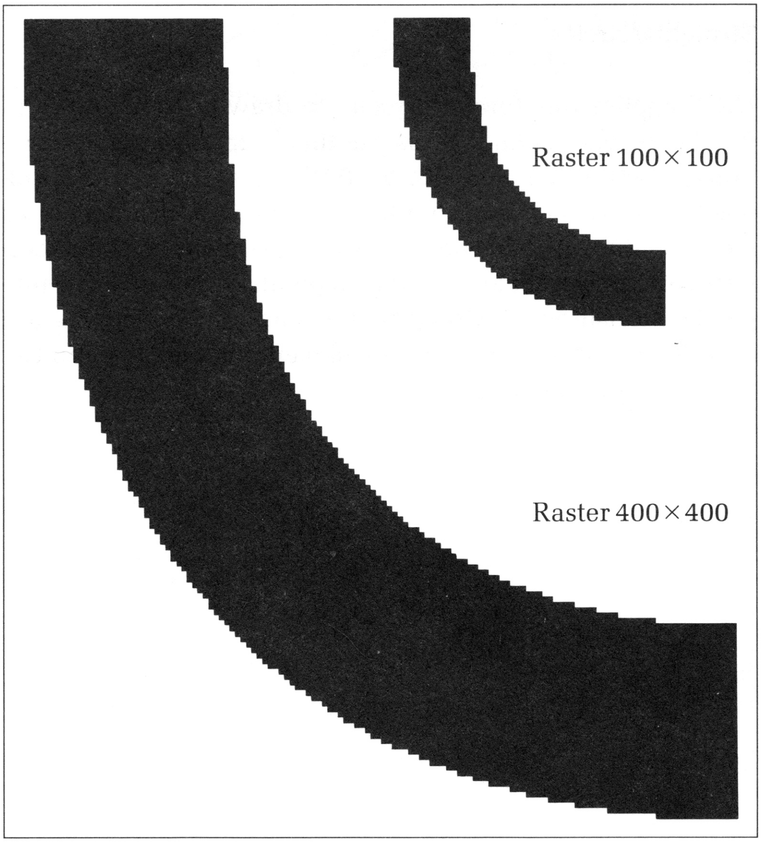


Fig. 53

5.1.5 Curved lines

«Curved lines» is a slightly inelegant term used to distinguish all non-linear, non-straight outlines. The term is used here to describe elements such as:

Math helps code curved lines

- Circles (Bitstream, URW),
- Bézier functions (PostScript),
- Spirals (Purdy, AM, Itek),
- Splines (XEROX and others)
- Conic splines (SUN/Imagen)
- Quadratic splines (Apple, Microsoft)

Straights/circles

URW applies this format directly to drawing machines and graphic displays. Other firms use this format for high resolution typesetting machines (60 to 100 lines/mm). It can be developed most directly and rapidly from the IKARUS format as mentioned later in this book. Basically, straight lines link corners with each other and to tangent points, where occurs a tangential transition of straights into curves. The curves are described as segments of circles all blending into one another tangentially (see also 6.3).

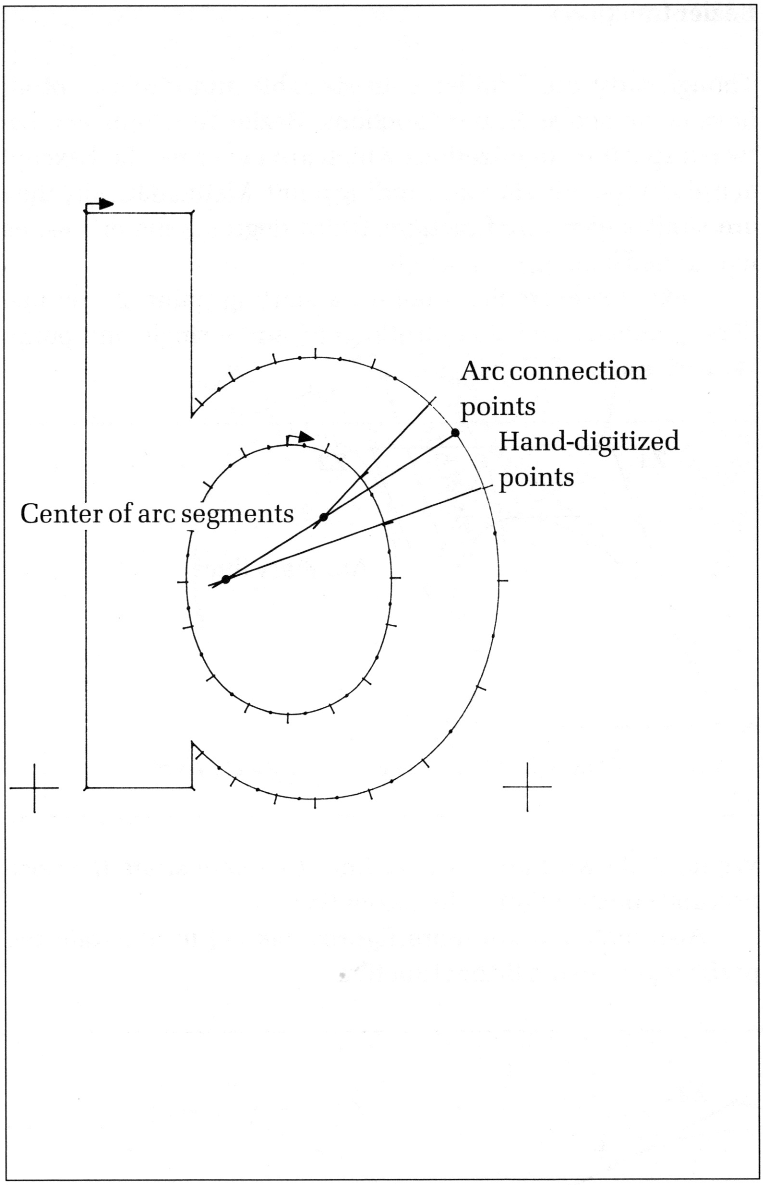

Arc connection points
Hand-digitized points
Center of arc segments

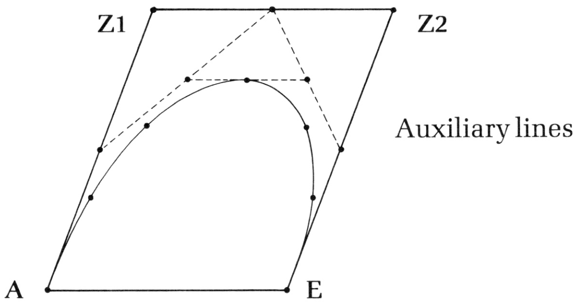
Fig. 54
DI format of lower case b at a raster of 15,000 × 15,000 for the em.
The combination of IKARUS and DI format is Bitstream format (each coarser raster is calculable).

Bézier functions

Cubic splines are defined by means of four points.

Though most are familiar with straights and circles, not all have come across Bézier functions. Bézier functions run between specified digitizations which are never reached except at a given specific start and ending point. Mathematically they are similar to spline functions (third degree polynomials, as mentioned later on).

Take, for example, 4 points; a starting point, 2 intermediate points (so-called control points) and a single end point. We arrive at the following:



$$B(x,y) = A(x,y) \times (1-t)^3 + Z_1(x,y) \times (1-t)^2 \times t + Z_2(x,y) \times (1-t) \times t^2 + E(x,y) \times t^3$$
$$t = 0,....,1$$

Fig. 55 The Bézier function requires two anchor and two control points.

We have shown thin auxiliary lines to demonstrate the position and construction of the Bézier line.

Also included are more figures (below) to illustrate the performance of the Bézier function.

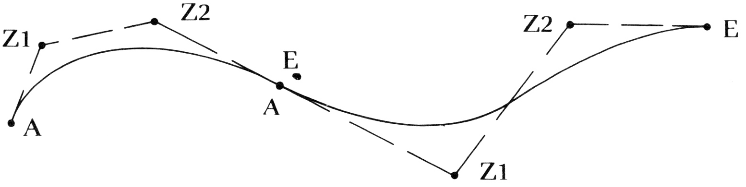


Fig. 56 Bézier representation of curves with inflection

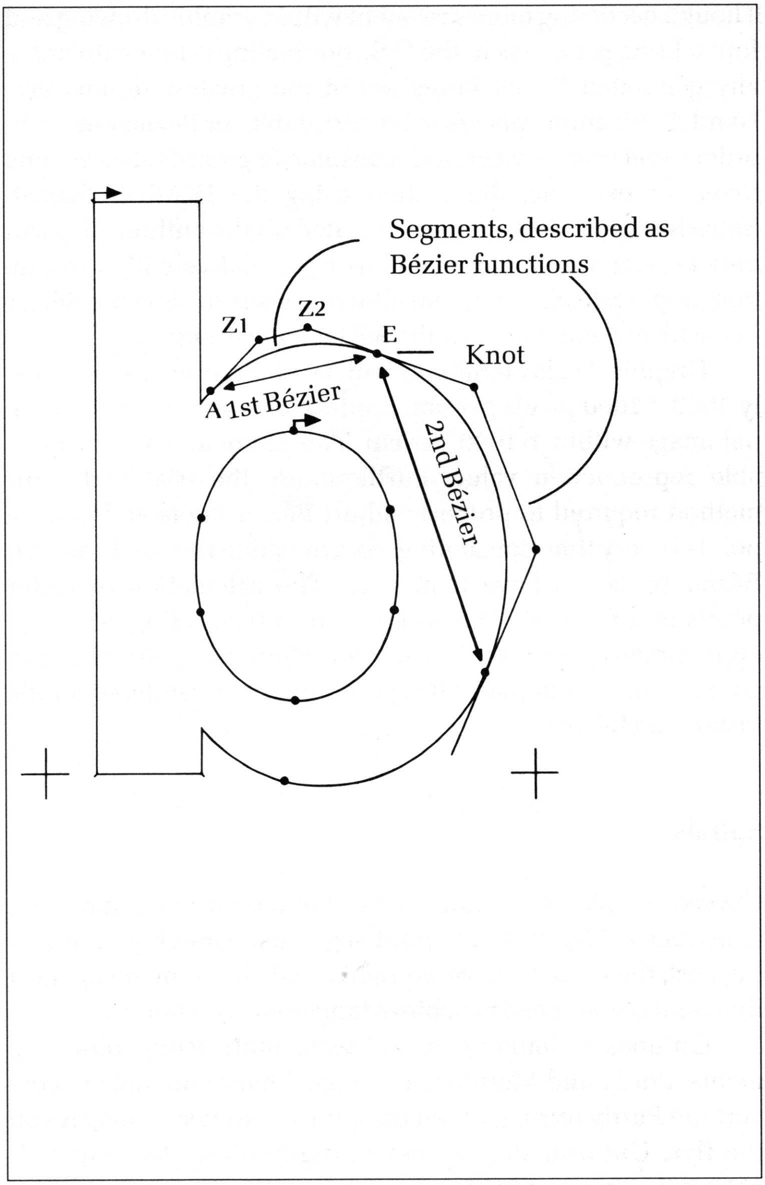

Segments, described as Bézier functions
Z1
Z2
E
Knot
A
1st Bézier
2nd Bézier

Fig. 57
PostScript format of lower case b at a raster of 15,000 × 15,000 for the em (each coarser raster is calculable).

Though becoming more prevalent within graphic drawing and font editing programs in the U.S., our feeling is that automatically generated Bézier knots are of the greatest serious use. Hand digitization appears to be unsuitable for Bézier curves in order to get precise work, as it consumes a great deal more time than, for example, digitization using the IKARUS format. Since Bézier control points are located off the outline, they can only be seen and manipulated using a relatively high resolution graphic display terminal allowing users to view the Bézier line with any movement of the Bézier control points.

IK format versus PostScript format

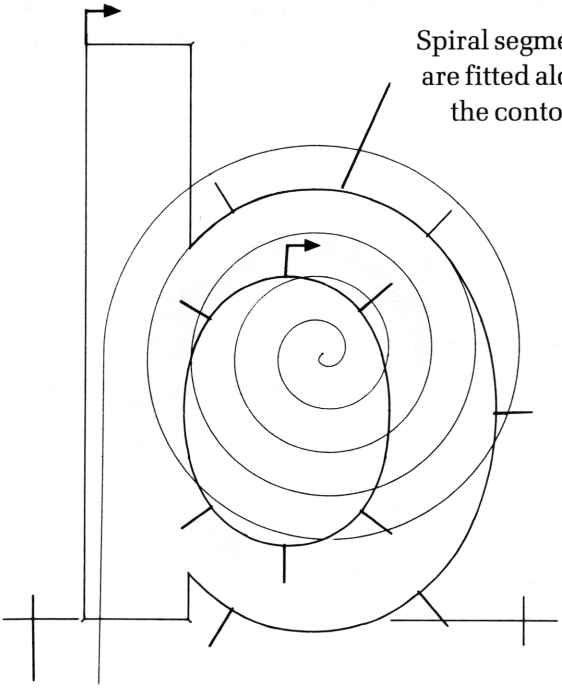
Graphic display terminals with resolutions of approximately 1000 × 1000 pixels per em, produce congruency to an original image within 1/1000 per em. This is not an easily achievable reproduction value. Furthermore, the trial and error method required to properly adjust Bézier knots and control points is very time consuming. As a machine format, however, Bézier functions have their uses. The calculation of raster points is sufficiently fast when used in typesetting, although approximately 6 times slower than when using straights and circles. Conic and quadratic splines could be rendered a little faster than Béziers.

Spirals

Curved templates are known as «French curves», and were constructed largely from spiral segments. Drawing letters in the past, the French curve, straights, and circles made up most lines and one scraped to achieve tangential transitions.

Spirals are also a suitable typeface coding method.

Outlines of letters can and were built using spiral segments. Purdy and McIntosh developed hardware able to convert the Purdy format, based on spirals, into raster images «on the fly». Unfortunately however, digitization (data input) is extremely time-consuming with a resulting resolution lower than 1/1000 unit per em. On average about 15 minutes interactive work is required on screen to fit spiral segments to a letter. In comparison, the average marking time per letter is only 2.6 minutes using the IKARUS format plus another 2.6 minutes of pure machine digitization time. For this reason mainly, we do not consider the Purdy format best suited to databasing characters. They are difficult to edit and not easily transformed or distorted.



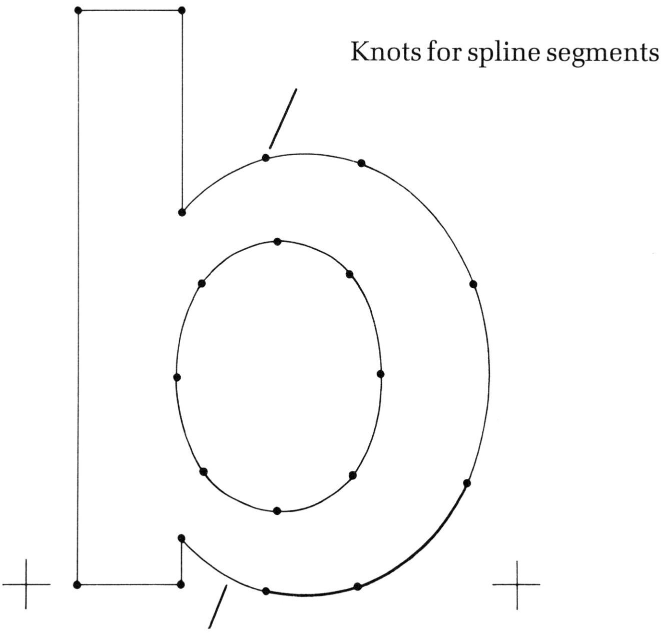
Spiral segments are fitted along the contour

Fig. 58
Purdy format of lower case b at a raster of 4000 × 4000 for the em (each coarser raster is calculable).

Splines

Spline interpolation?

Use of general spline functions to represent letters is a very good attempt in the right direction. We at URW used this method since 1972, but it is used less and less more recently. From the formater spline interpolation of IKARUS points today we use only the tangents of the digitized points. This is a



A segment is described by
$x(t) = a_x + b_x \times t + c_x \times t^2 + d_x \times t^3$
$y(t) = a_y + b_y \times t + c_y \times t^2 + d_y \times t^3$
t = line distance
$\{a_x ... d_y\}$ are constants for a segment

Fig. 59 Spline format of lower case b at a raster of 15,000× 15,000 for the em. The scanned knots are at a raster of 1000× 1000 for the em (each coarser raster is calculable).

To demonstrate in greater detail, the figure below shows the spline method for letters as used by XEROX (ref. Stone). It is apparent that automatic digitization is a good starting point; it would be very difficult and time-consuming if knots had to be located manually, guided by eye. However, this format has the disadvantage of not being editable by displacement of knots and therefore not transformable.

Auto-tracing?

Automatic digitization of letterforms within this format cannot reproduce the original more precisely than its scanning accuracy, approximately 1/1000 per em width.

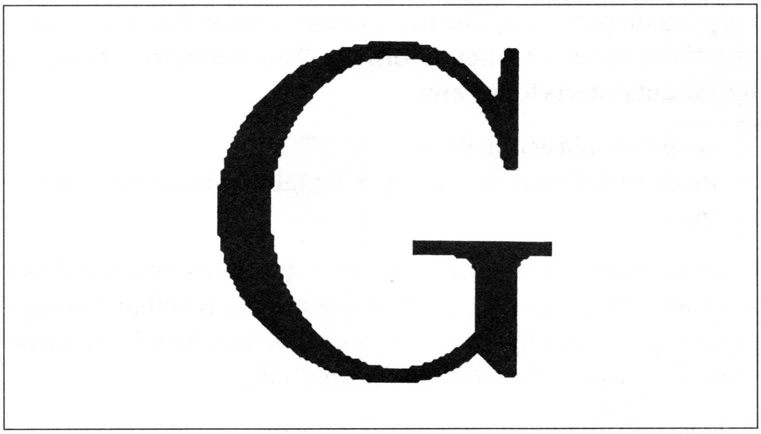


Fig. 60
Scanned original

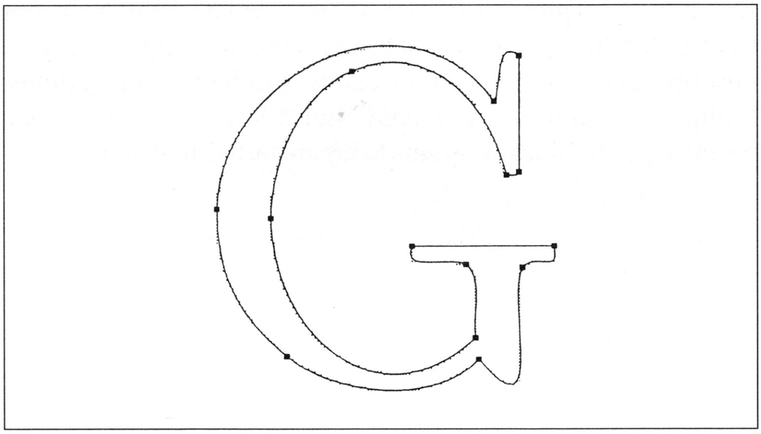


Fig. 61
Result of spline fit

5.2 Other concepts

A short history of earlier concepts for the preparation of digital types has been edited at Stanford University (ref. Ruggles). A concept is assessed in this study according to its performance in artificial generation of new letters, one might even say on the degree of complexity envisioned by respective inventors.

While not wishing to reject such criteria for assessment, we cannot ignore the fact that the major criterion has been completely omitted.

There are thousands of typefaces available today for all imaginable purposes. The main consideration for us is to convert these faces into digital formats. This being true, the most important criteria today are:

- speed of data acquisition
- quality of storage (accuracy, adaptability into machine formats).

It is not the new typefaces that make the work, but the existing ones and their integration into today's formats and platforms.

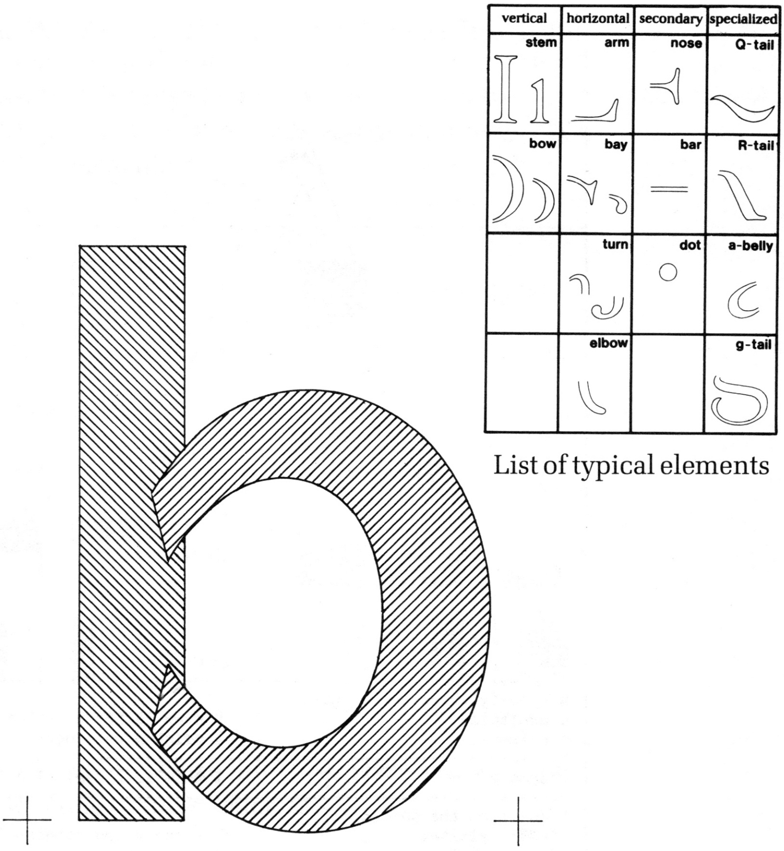
Computer-aided generation of new types must be considered as of secondary importance. The question is, whether a designer, using a computer to accomplish this task, would be faster than using conventional drawing methods.

At least two other concepts are of interest:

In contrast to previously mentioned formats, which do not treat individual letters as such, but use a more global digital description of a font as either pixels, run lengths or outlines, Philip Coueignoux and Donald Knuth have created formats requiring individual programs to create individual letters.

Philip Coueignoux divided letters into elements and then designed appropriate software and hardware to generate letters from these elements. The finished type reproduction – considering the larger cross section of type – does, however, leave something to be desired. The advantages of the Coueignoux format are its small storage requirement.

Fig.62
Latin fonts comprise only a few elements.


vertical
horizontal
secondary
specialized
stem
arm
nose
Q-tail
bow
bay
bar
R-tail
turn
dot
a-belly
elbow
g-tail

List of typical elements

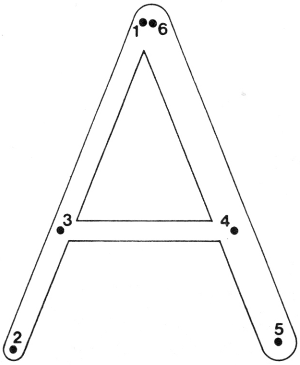
1st element for b,d,h,k,l,(t)
2nd element for b,p

The Coueignoux format consists of typeface elements and individual software to cut and to form characters (each coarser raster is calculable).

Metafont is too slow for type production.

Donald Knuth created the Metafont concept, very well suited to modifying a Metafont typeface. While modification possibilities are virtually endless, Metafont is not well suited for exact and cost effective reproduction of existing typestyles. A further problem is that a rather long program must be written for each individual character.

Steps have been taken since 1983 to allow hand-digitized data to be used also as descriptive data.



```
% The letter A
%
%----------------------------------------------------------------
% specify where output goes
drawdisplay;                      % draw letter on screen
proofmode;                        % print proof sheet
%----------------------------------------------------------------
charcode 'A;                      % this is a capital A
%----------------------------------------------------------------
% position the points
x1=98;  y1=250;                   % x and y coordinates for each of
x2=0;   y2=0;                     % 6 points
x3=40;  y3=90;
x4=170; y4=y3;
x5-105; y5=y1;
x6=203; y6=5;
%----------------------------------------------------------------
% specify the pen
cpen;                             % circular pen nib
%----------------------------------------------------------------
% draw the charcter               % using a pen width of 15, draw a line
15 draw 1..2;                     % between points 1 and 2, and between
   draw 3..4;                     % points 3 and 4; then with a pen
25 draw 5..6;                     % of with 25, draw a line between
                                  % points 5 and 6.
end
```

Fig. 63
The Metafont program for an upper case A. All raster resolutions are programmable (each coarser raster is calculable).

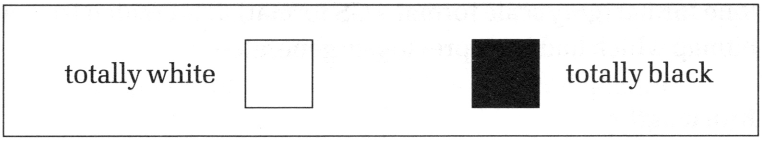
5.3 Special coding techniques

Over the past nineteen years various special coding techniques have been developed. Some of them are described in the following section.

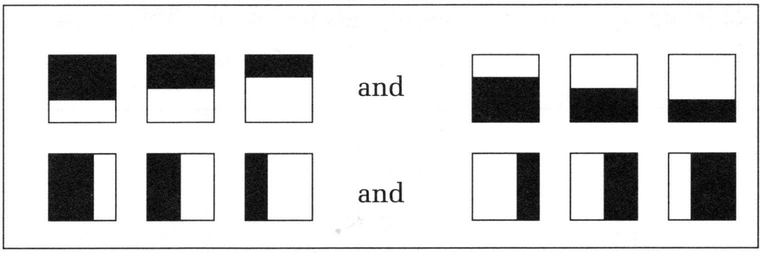
Bitmaps

Block encoding

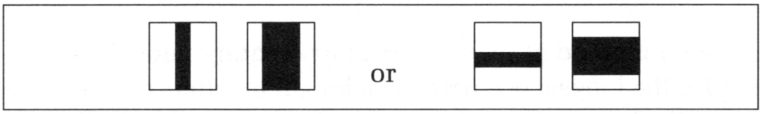
Bitmaps could also be coded as block encoding (ref. Knudson). As an example, for 4 × 4 bits it is not necessary to use 16 bit data, since many of the approximately 65,000 combinations do not appear within the black and white image of a letter. We definitely have the elements:



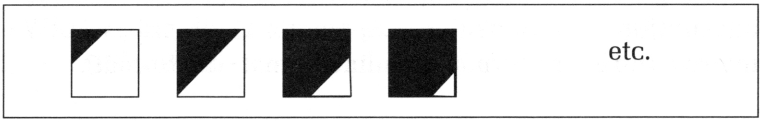
and the element groups:



If enough information on stroke thickness were available, one could perhaps exclude thinlines such as:



A similar argument applies to diagonals which perhaps only appear in the following forms:



One can almost sense that the reproduction of letters may very well be achieved with only $2^8 = 256$ possibilities, a reduction of necessary storage capacity by a factor of 2. In Appendix Q we describe our own internal bitmap format (BI format).

Bytemaps

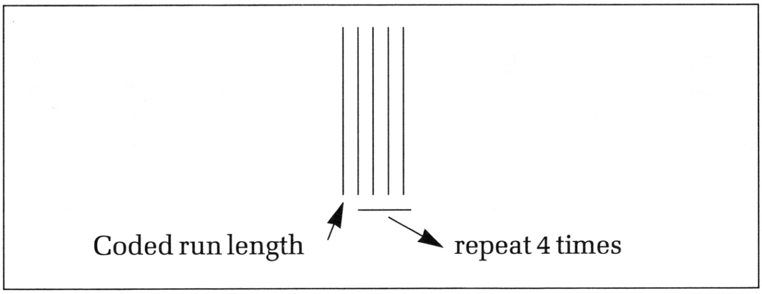
In halftone (grayscale) pictures it is relatively rare – unlike text – to see extreme black/white transitions. The extremely compact storage of halftone pictures when used in telecommunications makes use of this fact,with an average of 2 bits per gray pixel (normally 8 bits) to code relative color tone change of one image point to that of its predecessor.

In Appendix R we have described our own internal halftone format (gray scale format – GS format). It is created from a bitmap which had been previously generated.

Run lengths

Run length coding can compress a bitmap to a tenth of the original.

Various different run length codes have already been considered. As a rule, a compression of approximately 30% may be achieved by using the «line repeat» technique. This shows an «I»:



Another method of reducing memory requirements is searching for the longest occurring run length in letters or type for a prescribed raster. From this the n-bit-byte is calculated, allowing this run length as well as all shorter run lengths to be coded incrementally. «Incremental» refers to data concerning respective lengths of black or white line segments, rather than absolute positions of transitions. Appendix O describes URW's internal run length format (scanline format – SC format).

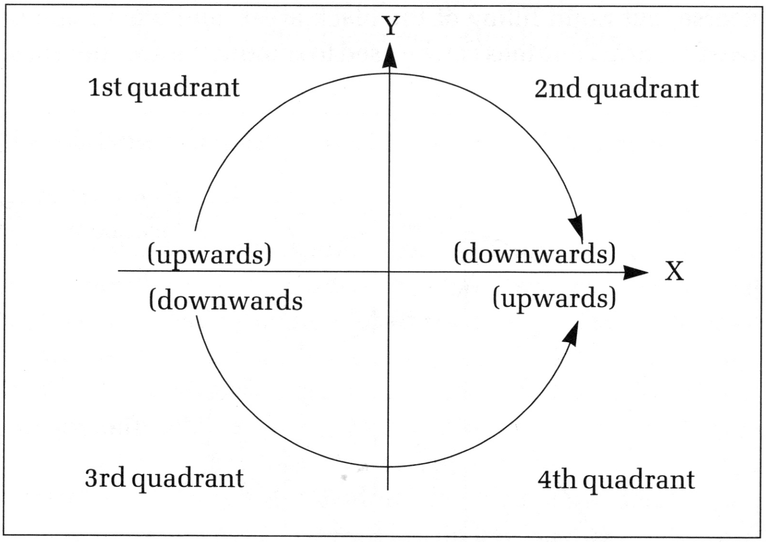
Vector formats

Open vector formats, running in short steps from edge point to edge point, are available. Coding of this kind is known as «nibble» coding. We have concluded that two economical possibilities exist:

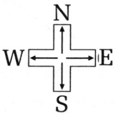
Nibbles and 2-bit chains

a) 1-bit-nibbles	0	straight in X direction →
	1	straight and vertical skip ↗

This method requires super-imposing coding of the appropriate quadrant, e.g., when describing a circle. For certain quadrants «1» can only mean «upwards,» for others it can mean «downwards».


Y
X
1st quadrant
2nd quadrant
3rd quadrant
4th quadrant
(upwards)
(downwards)
(downwards
(upwards)

b) 2-Bit-Nibbles	0	≙	North
	1	≙	West
	2	≙	South
	3	≙	East



Using this code it is possible to completely describe circles or spirals, while coding remains very compact, allowing segments to be accessed by being addressed with pointers from/to.

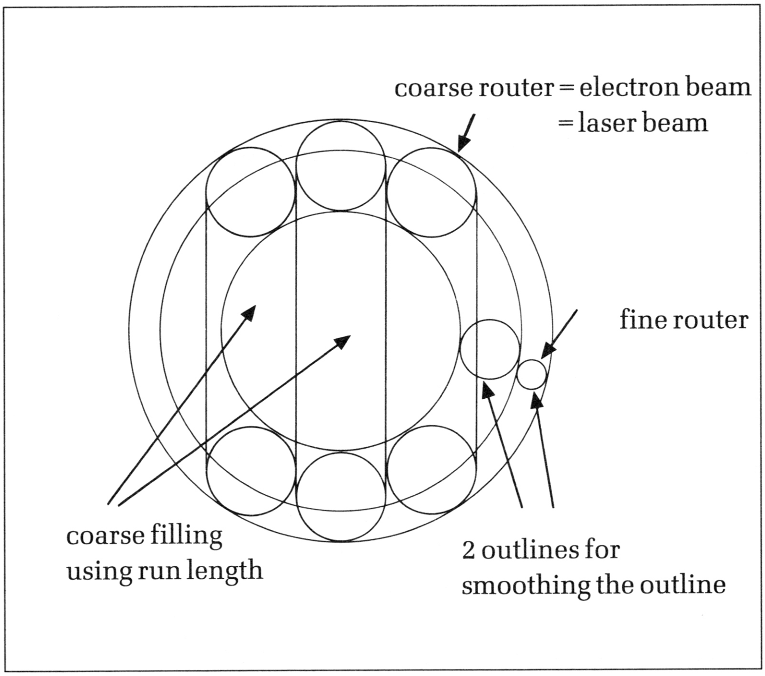
See Appendix N for a description of our internal vector format (VE format).

Complex outlines and run lengths

Using the chord length instead of the radius to describe a circle.

Though some exist currently, in the future, we should expect to see more machines able to directly handle and process complex outlines. With a technique corresponding to the vector format straights can be coded as usual, and circles incrementally. For coding circle segments the end point is stated as a vector plus chord height. An advantage in this is elimination of the problems of coding large radii, as 3 bytes per circle segment are sufficient. A shortcoming is that the output of the chord height requires greater computer capacity to regenerate circle data on a display. See Appendix L for a description of our VC format.

We are not aware of any typesetting machine in which run lengths and outlines are so interrelated that run lengths lead to coarse, but rapid filling of the black areas, and where one or two fine exact outlines can be used to smooth the area outlines.


coarse router = electron beam
= laser beam
fine router
coarse filling
using run length
2 outlines for
smoothing the outline

It is likely that this method will be utilized in future routing and other engraving machines.

General comments

The cost of data storage and electronic equipment itself continues to fall, which will have these positive consequences concerning production of digital type formats:

Digital typefaces are an upward trend.

- Concentration of effort on coding and compression of data will no longer be necessary.
- Digital storage of fine rasters instead of coarse rasters will become practicable.
- Storage of complex curves will be easier and faster, and will therefore become a workable reality.
- Additional typesetting data, such as kerning and touching tables, are more readily stored and processed.
- More typefaces will be available per system.
- More characters will be accessable per typeface.

5.4 Listing of a selection of type manufacturers

In the following tables we have used the designation «vector format» describing machines from different manufacturers. This should not be construed to mean, however, that all machines use identical coding for the same digital type. Vectors, as we have seen, may be coded in «nibbles», open, closed, incremental, absolute, normal, rotated, or perhaps mirror-image coordinate systems. Many different and varied compression techniques may also be applied (vector repeat, sign change, etc.). The field of coding allows so many possibilities that, to our knowledge, regrettably no two manufacturers use the same coding method if they have invented it by themselves.

Company	Country	Machine/ product	System used	Kind of data base	Kind of machine format
Adobe	USA	Software, PostScript	own	hand-digitized IKARUS data	curved lines, bitmaps,
AGFA Compugraphic	Belgium/USA	laser printers, var. setting machines	own/IKARUS Intellifont	hand-digitized	bitmaps, curved-lines, vector format
Alphatype	USA	CRT-machines	own	scan data	open vector format
ARISTO	Germany	drafting-tables	–	–	curved lines
Artronics	USA	image makeup	–	IKARUS data	halftone
Asahi	Japan	newspaper	IKARUS	scan data/ hand-digitized	curved lines
Autologic	USA	CRT-machines	own/IKARUS	scan data/ hand-digitized	curves, nibbles
BBC	UK	TV	own/IKARUS	IKARUS-data	halftone
Berthold	Germany	various setting machines	IKARUS	hand-digitized	curved lines, vektor lines
Bigelow & Holmes	USA	typefaces	IKARUS	hand-digitized	all formats
Bitstream	USA	typefaces	LIP	hand-digitized	Bitstream-format
Brother	Japan	laser printer	IKARUS	scan data/	
CGK	Germany	laser setting machine	own/IKARUS	hand-digitized	curved lines
Crossfield	UK	page makeup	–	–	halftone
Dainippon Printing	Japan	setting-system, printing	IKARUS	scan data/ hand-digitized	curved lines
Dainippon Screen	Japan	page makeup	IKARUS	hand-digitized	curved lines
Fujitsu	Japan	laser printer	IKARUS	scan data/ hand-digitized	curved lines
Gerber	USA	sign makers	own	hand-digitized	curved lines
Hewlett Packard	USA	laser printer	Symbolics intellifont	scan data/ hand-digitized	curved lines
Honeywell Bull	France	laser printer	own	–	bitmaps
HTS	USA	setting system	–	–	

Company	Country	Machine/ product	System used	Kind of data base	Kind of machine format
IBM	USA	laser printer	own/IKARUS	scan data/ hand-digitized	bitmaps curved lines
III	USA	CRT-machines	own/IKARUS	scan data/ hand-digitized	run lengths, vector lines
Imagen	USA	laser printer	own	scan data	bitmaps
ITC	USA	typefaces	–	hand-digitized IKARUS-data	analog
Itek	USA	CRT-machines	own	scan data/edit	vector format
Iwata	Japan	typefaces	IKARUS	scan data hand-digitized	curved lines
Letraset	UK	transfer-letters	IKARUS	hand-digitized	analog
Letterphot	Germany	photosetting machines	analog	–	analog
Linotype-Hell	Germany	setting machines page makeup	own/IKARUS Scriptus filmrecorders	scan data/ hand-digitized	curved lines halftone
Mainichi	Japan	newspapers	IKARUS	hand-digitized scan data	curved lines
Mannesmann-Scangraphic	Germany	CRT-machines	IKARUS	scan data/ hand-digitized	curved lines
Matsushita	Japan	laser printer	own	bitmaps	bitmaps
Monotype	UK	setting machines	own/IKARUS	scan data hand-digitized	run lengths, curved lines
Morisawa	Japan	laser records	own/IKARUS	scan data/ hand-digitized	curved lines
Moskau I	UdSSR	–	IKARUS	hand-digitized	–
Moskau II	UdSSR	–	IKARUS	hand-digitized	–
Motoya	Japan	typefaces	own/IKARUS	scan data	vector formats
NEC	Japan	various printers	own/IKARUS	hand-digitized/ scan data	vector formats
Nikkan Sport Newspaper	Japan	newspapers	IKARUS	hand-digitized/ scan data	vector formats
OCE	Netherland	laser printer	own	bitmaps	bitmaps

Company	Country	Machine/ product	System used	Kind of data base	Kind of machine format
OEL	Japan	setting machines	own/IKARUS	scan data/ hand-digitized	curved lines, run lengths
Olympia	Germany	laser printer	own/IKARUS	bitmaps	bitmaps
Philips	Netherland	laser printer	own	bitmaps	bitmaps
PMT	Japan	CRT-machines	IKARUS	hand-digitized	vector format
Purup	Danmark	laser recorders	own/IKARUS	hand-digitized	IKARUS-data, vector data
QMS	USA	laser printer	own	bitmaps	curved lines
Quantel	UK	image makeup	own	hand-digitized IKARUS data	halftone
Qume	USA	laser printer	IKARUS	bitmaps	bitmaps
Ryobi	Japan	setting machines	own/IKARUS	scan data/ hand-digitized	curved lines
Scitex	Israel	page makeup	own	hand-digitized	Bitstream-format
Sha-ken	Japan	setting machines	own/IKARUS	scan data/ hand-digitized	curved lines
Sharp	Japan	laser printer	own/IKARUS	scan data	vector formats
SIL (Shanghai-Ikarus LTD)	China	Kanji	IKARUS	hand-digitized	all formats
Siemens	Germany	laser printer	own/IKARUS	hand-digitized	bitmap
Signus Ltd.	UK	typefaces	IKARUS	hand-digitized	all formats
Tegra-Varityper	USA	CRT-machines	Purdy	scan data	Purdy-format
Tektronix	USA	VDTs	–	–	vector data
Toppan	Japan	setting-system	IKARUS	hand-digitized	vector format
Toray	Japan	laser printer	IKARUS	scan data/ hand-digitized	vector format
Typebank	Japan	typefaces	own/IKARUS	hand-digitized	all formats
URW	Germany	typefaces	IKARUS	hand-digitized	all formats
Versatec	USA	electrostatic printer	–	–	vector data, bitmaps
Visual Graphics	USA	photosetting-machines	analog	–	analog
Wild	Switzerland	drafting tables	–	–	curved lines
XEROX	USA	laser printer	own/IKARUS	scan data/ hand-digitized	bitmaps, curved lines

The URW concept

The backbone of the IKARUS concept is storage of hand-digitizations from which all machine formats may be calculated. This concept – storage of hand-digitizations – is being used currently by manufacturers of typesetting machines.

6.1 IKARUS

Digital formats, for the most part, are utilized to display characters on machines. The faster and more powerful the microelectronics in a machine, the more complex the format may become.

Considering nothing but the machine, it is possible to overlook the digital form itself.

Hand-digitization is a true digital form. It can be accomplished extremely accurately and quite quickly, about 2.6 minutes per character attaining 1/100 mm accuracy at a cap height of 10 cm. The em raster is 15,000 × 15,000 pixels in size. Such fine resolution permits use of not only black and white original artwork, but even pencil sketches or poor copies of art, with the human brain itself smoothing any roughness from an original in the process of digitizing. In Appendix X rules for marking and digitization of characters is shown.

IKARUS stores the em at a raster of 15,000 × 15,000 units.

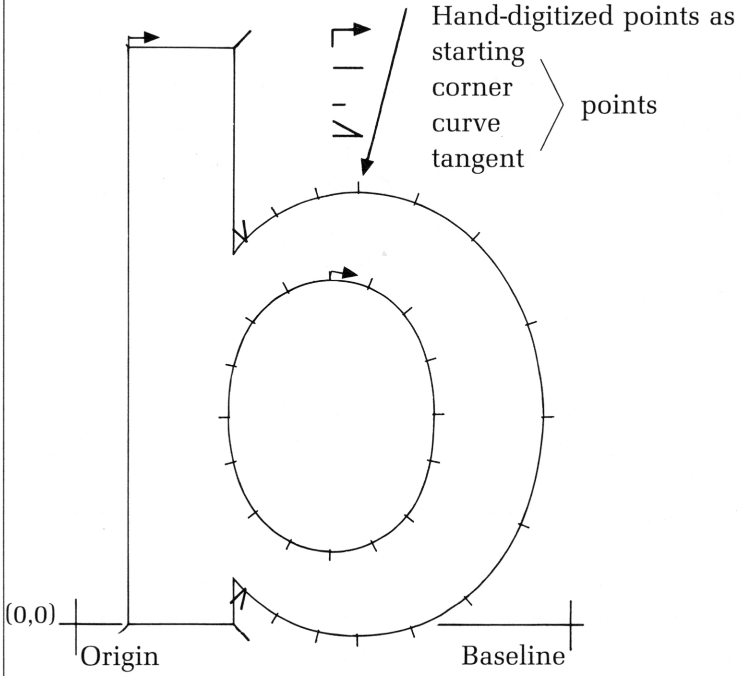

Hand-digitized points as
starting
corner
curve
tangent
points
(0,0)
Origin
Baseline

Fig. 64
IKARUS format of lower case b at a raster of 15,000 × 15,000 for the em (each coarser resolution is calculable).

The digitization points placed by the operator specify the run of letter outlines. For this reason alone, hand-digitization creates the most critical, non-transient digital data forming the basis of the entire operation (see Appendix H for a description of the related IK format). A requisite consideration when digitizing is that other digital formats may be derived from the information. This has proven to be no problem to date, as the hand-digitized data was sufficient for all necessary calculations.

Bitstream Corporation has adopted the IKARUS concept for their use. They hand-digitize and store as at URW, however the Bitstream format also combines a complex outline description of straights and circles with hand-digitization.

6.2 Format hierarchy at URW

URW formats are open.

URW has developed internal formats having particular relationships to one another.

IK format	(IKARUS , hand-digitizations, appendix H);
BE format*	(Bézier curves, appendix I);
CN format*	(Conic-curves, appendix I);
QQ format*	(Quadratic-splines, appendix I);
II format	(IKARUS with instructions, appendix J);
DI format	(display, circles/straights, appendix K);
VC format	(vector/circle, appendix L);
BS format	(Bézier with instructions, appendix M);
VE format	(vectors, appendix N);
SC format	(scanlines, run lengths, appendix O);
SN format	(scanline nibbles, appendix P);
BI format	(bitmap, appendix Q);
GS format	(grayscale, bytemap, appendix R).

Formats are detailed in the appendices, written, in part, in a non-compressed form. This was done so respective data users may prepare from transferred data their own compression and safety encryptions matching their own electronics. We have so far been successful in using our own non-compressed formats to calculate all existing digital machine formats for clients. For example, the internal VE format was used to create special vector formats (programming the post processor) for a client in a period of one day to six weeks, averaging two weeks.

---

* Formats without instructions (i.e. without hints)

A deeper hierarchical step may be reached using programmed conversions. URW uses such version of:

IK	⇒	DI	program DI
IK	⇒	VC	program VC
DI	⇒	SC	program SC
SC	⇒	BI	program BI
BI	⇒	GS	program GS

The reverse manipulation, is not so straightforward, and theoretically should not be necessary, as it is always possible to use the hand-digitization for reference. However, companies have invested more than 100 man-years in preparing a digital type library based on scanline data. To utilize this information, these companies must use their data to create an outline database form to meet potential requirements of future machines. The problem might be solved in this manner:

LINUS

Auto-tracing

The LINUS system consists of the necessary software for conversion of scanned to digitized data. This software utility was basically designed for reading and converting black and white pictures. Recently, the ability of processing coloured images was added. The resulting data is compatible with hand-digitized data and is of nearly the same accuracy. Below you can see an example of both hand and automatically generated outline digitization.

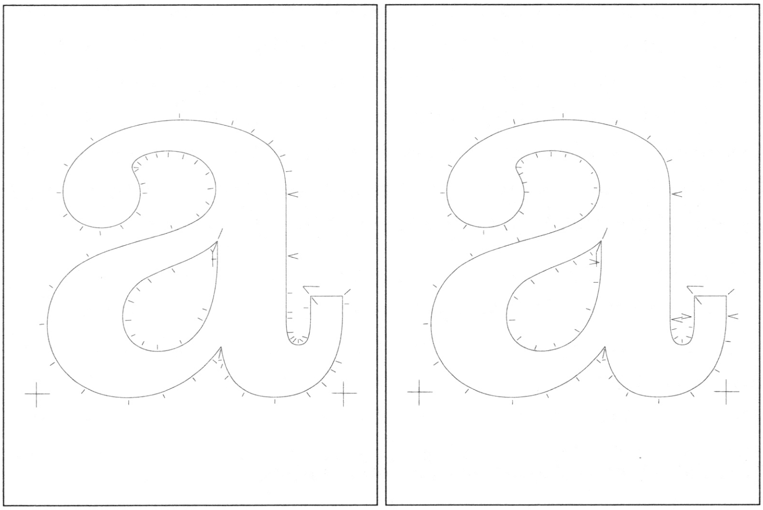


Fig. 65 (left side) hand-digitized (right side) automatically digitized using scan data

Reverse conversions (VE → IK) at URW required a substantial effort (1979 to 1984). Other reverse conversions, however, were much simpler to program.

DI ⇒ IK no problem
SC ⇒ VE no problem
BI ⇒ SC no problem
GS ⇒ BI is not possible, but then again not required.

Inherent in the hierarchy, and fundamental in URW philosophy, is the fact that the IK format incorporates a precision greater than that required by typesetting machines. Characters in this format are stored within a raster of 15,000 × 15,000 units per em. A distinct advantage of this resolution is in preparation of coarser rasters there is no interference or conflict between the stored and target rasters.

Typefaces are expected to give years of service, so it is worth recording them with highest precision.

The finest raster required by most typesetting equipment is 2,500 × 2,500 units per em, 400 × 400 units per em for «good» typesetters and other machines even coarser than 400. IKARUS rasters are usually a factor of 10 finer than the output device which means that both a 400 × 400 and a 401 × 401 unit raster can be calculated to the same quality.

The 400 lines may be placed precisely on the IKARUS raster with about 1/40 line separation, likewise the 401 lines may be placed with slightly less separation between, namely:

at 400 with 15,000/400 = 37.5
at 401 with 15,000/401 = 37.4

The line separation resulting is:

37, 38, 37, 38, 37, 38, 37, 38, ...
and
37, 38, 37, 38, 37, 37, 38, 37, 38, ...,
as counted in IKARUS fine raster points.

The earlier reference to «no interference between storage and target raster» should be more understandable at this point. It would be different if the storage raster were just 800 × 800 units per em. Although a 400 × 400 raster would present no problem, merely accessing every other pixel and using every second line, a raster of 401 × 401 would be much more complex. We would have 398 times a line separation of 2 and twice a line separation of 1, counted in storage raster points. It would be necessary to acquire 2 sequential lines from the 800 one time, or continuously interpolate a new line between 2 lines correlated to their real position between these lines. Either method produces errors! Fortunately, the fine resolution possible in IKARUS data is a good solution to these problems and no errors are encountered.

Fineness of the digitized data in the «digitized once and for all» IKARUS mode is a given advantage, but additionally, storage of the actual hand-digitizations is of extreme importance. Only using such data can subsequent data manipulation, like moving a single point (corrections) or calculating a spheric projection (typesetting on a three-dimensional globe) be easily carried out.

All IK points are «on-curve».

The manner and method of digitization is summarized in Appendix X. Manipulations are consistently followed by calculation of a closed curve through a given hand-digitization, and are regarded as «knot» points.

In choosing this procedure, a great deal of time was invested determining the cleanest and most elegant technical route. It is of primary importance that curve calculations are carried out so that each hand-digitized point coincides with the calculated line.

For this reason, we have found that Bézier functions are less desirable in type digitization of existing typefaces. They do not hit the original points. For the same reasons we have also not included a curve fit calculation, only the curve interpolation is taken into consideration.

6.3 IK interpolation

Spline interpolation appears to be the best solution for calculation of the tangent directions of digitized points. Calculations were also carried out using circle fitting, polynom fits as well as other methods, they were compared to spline interpolation, and we rejected them all.

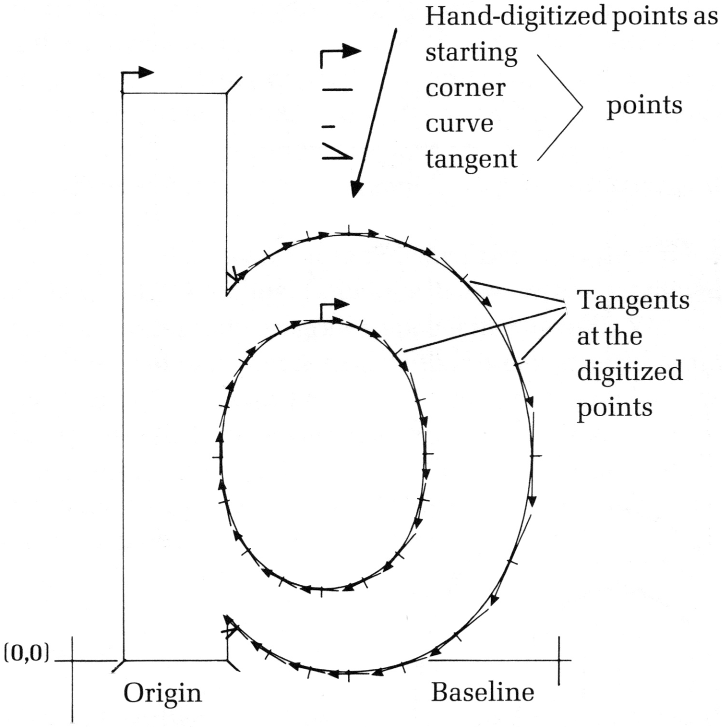

Hand-digitized points as
starting
corner
curve
tangent
points
Tangents at the digitized points
(0,0)
Origin
Baseline

Fig. 66
IK format with tangents drawn at the digitized points.

The step taken following tangent calculation might seem at first to be unusual. The spline function is not taken to display the closed curve, but rather, two circle segments are used between any two digitized points. The first segment starts at the beginning point with the first tangent determined, stops midway having a specific direction. The second segment begins at mid-point, in the same direction, and stops at the second digitized point, determining the second tangent. Thus, a curve is

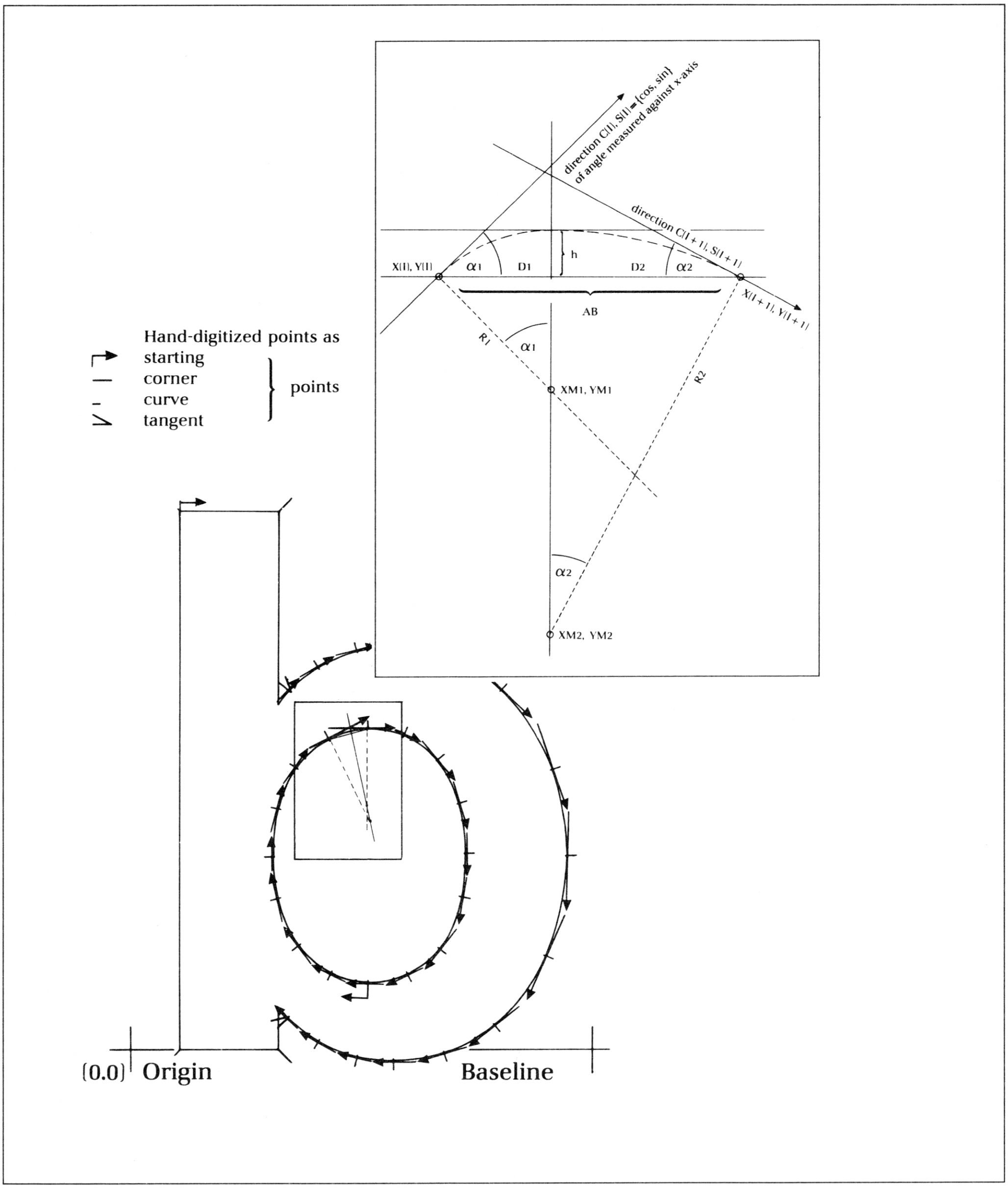

direction C(I), S(I) = (cos, sin) of angle measured against x-axis
direction C(I + 1), S(I + 1)
X(I), Y(I)
α1
D1
h
D2
α2
AB
X(I + 1), Y(I + 1)
R1
α1
XM1, YM1
R2
α2
XM2, YM2
Hand-digitized points as
starting
corner
curve
tangent
points
(0.0) Origin
Baseline

Fig. 67
Two arc segments can be determined on curves between each two IK points. These arc segments merge tangentially.

generated by linking a chain of arc segments tangentially merging into one another. Graphically, this method has the advantage of smoothing the curve path in steps. Even further, straights and circles allow very fast calculation as to whether or not any point is on the line itself. This latter feature is very useful in enabling a series of control calculations, such as margin justification, clipping, letter touching or layering.

If that were not enough, an added benefit of this method is that it does not produce undesireable variations of curvature, requiring the inclusion of damping terms in spline functions for their exclusion. Introduction of damping terms complicates the conversion to control signals for outline descriptions «on the fly» generating character output during typesetting. Straights and circles do not present the same problems. Electronic modules are currently available to, assisted by a DDA (Digital Differential Analyzer), to rapidly convert straights and circles into pixels.

Circle parameters are more quickly calculated than the 8 coefficients of the spline functions between two digitized points. The appropriate programs, appear in Appendix W.

The result of these curve calculations is called a DI format (display format) and can be applied directly in graphic displays, in some typesetting machines and on all NC machines.

This display format is used in URW's SIGNUS-system, very popular in the sign making industry for production of lettering for signage.

6.4 Nimbus

Intelligent font-scaling

Nimbus is a software package used in desktop publishing. It converts an outline font into a bitmap at any size from 4 to 270 point, at any resolution from 240 to 800 dpi (dots per inch), and meets demands of current laser and dot-matrix printers. NIMBUS can be used on a PC or in a raster image processor (RIP) in an interface for laser printers or high resolution screen displays. It can also serve as a generator for outlines in graphic art applications.

6.4.1 Core-fonts

In 1980, at the outset of desktop publishing (DTP), all laser printer manufacturers stored their fonts using bitmaps as their digital form.

Although bitmaps can be tuned manually using a bitmap editor, improving quality somewhat, storage of fonts as bitmaps for a complete range of sizes requires extensive amounts of memory, an expensive commodity, especially at that time.

Storage requirements of a single font style or design (4 variations: normal, italic, bold, bold italic) in 4, 6, 8, 10, 12, 14, 16, 20, 24, 36 and 72 pointsizes are around 2 Mbytes for a mixed bitmap/run length format. Storing but 13 core-fonts on a normal floppy disk, it was necessary to devise a more efficient storage method. In 1983, Adobe Systems and Apple Computer started collaborating on a raster image processor (RIP) to carry out intelligent font-scaling operating at printer resolution, using the outline description as the source font in order to generate many different pointsizes at low resolutions.

Adobe's concept quickly has become a success in the expanding field of desktop publishing. Typical memory requirement for a single typeface stored in outline format is between 30 and 50 Kbytes, allowing all 13 core-fonts to be stored on a single floppy disk.

NIMBUS Roman

NIMBUS Roman bold

NIMBUS Roman italic

NIMBUS Roman bold italic

NIMBUS Sans

NIMBUS Sans bold

NIMBUS Sans italic

NIMBUS Sans bold italic

Courier

Courier bold

Courier oblique

Courier bold oblique

Symbol set

♥ ♦ ♣ ♠

Fig. 68
There are thirteen core-fonts. Nimbus-Roman is the URW equivalent of the Times, the best known typeface family of the Monotype Corp.

Nimbus Sans is the URW equivalent of Helvetica, the best known typeface family of Linotype.

Nimbus Roman and Nimbus Sans have width tables based on the adaptions of Adobe Systems Inc.

At URW, the Courier typeface is shipped under the name Nimbus Mono.

6.4.2 Hard scanning

URW offers a number of outline font formats as a basis for NIMBUS in addition to BS-format (also II-format), T1-format, TrueType and Intellifont.

NIMBUS rasterizes most rapidly and efficiently using a vector/circle outline. Presently an average of 90 characters per second (cps) is achieved on an IBM PS/2 (386).

Soft-scanning might be best described thus: A program lays a raster or grid (its size elements based upon the pointsize chosen) beneath an outline character shape, and proceeds to search for points of intersection between the grid above and the outline letter below.

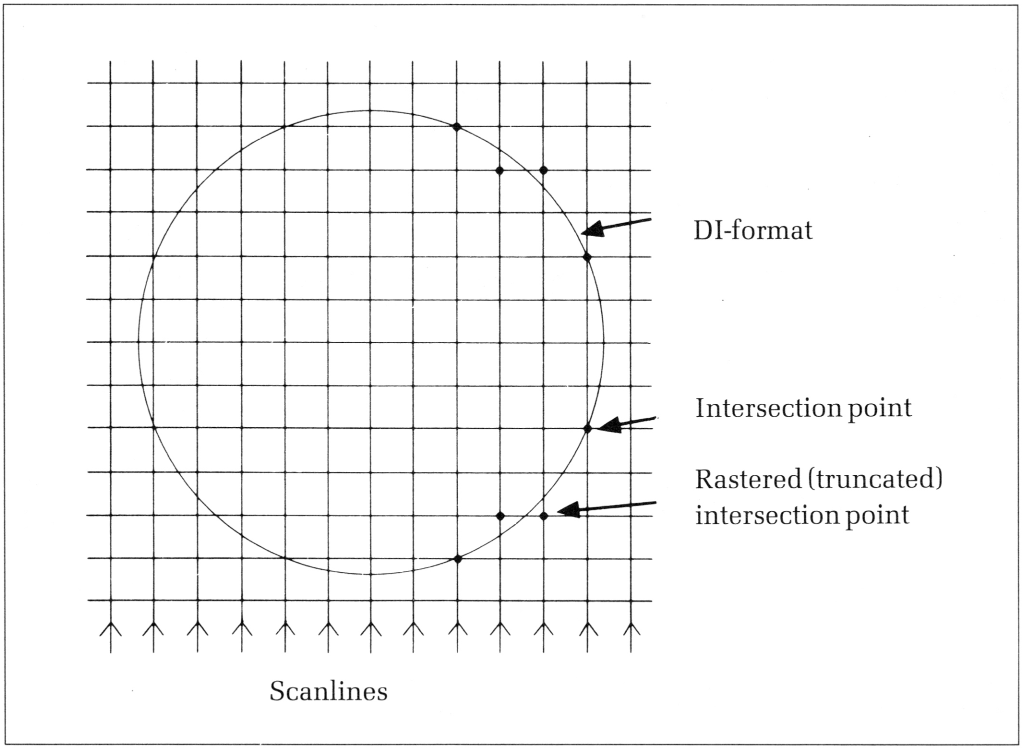
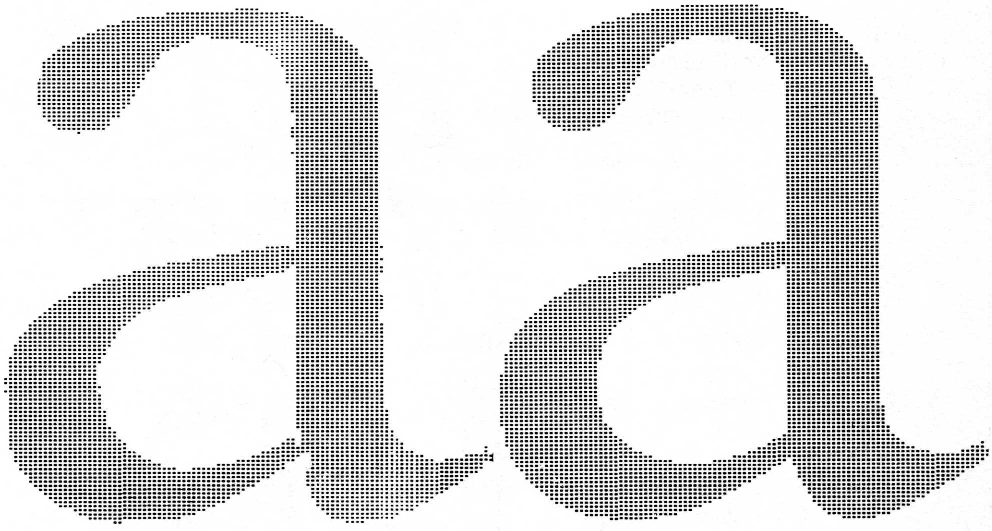


Fig. 69
Soft-scanning allows the application of any size of grid, whether fine or coarse, to the outline form.

Soft-scanning produces images significantly differing from those produced by an electronic scanner, or «hard-scanner.»

There are advantages in soft-scanning over hard-scanning, such as:

- The process is accomplished with software inside a computer without manual intervention.
- Dirt smudges, ragged edges and other visual interference are not a problem. They cannot occur.
- Errors and inconsistencies of black/white edges, introduced by electronic noise, are obviously avoided.

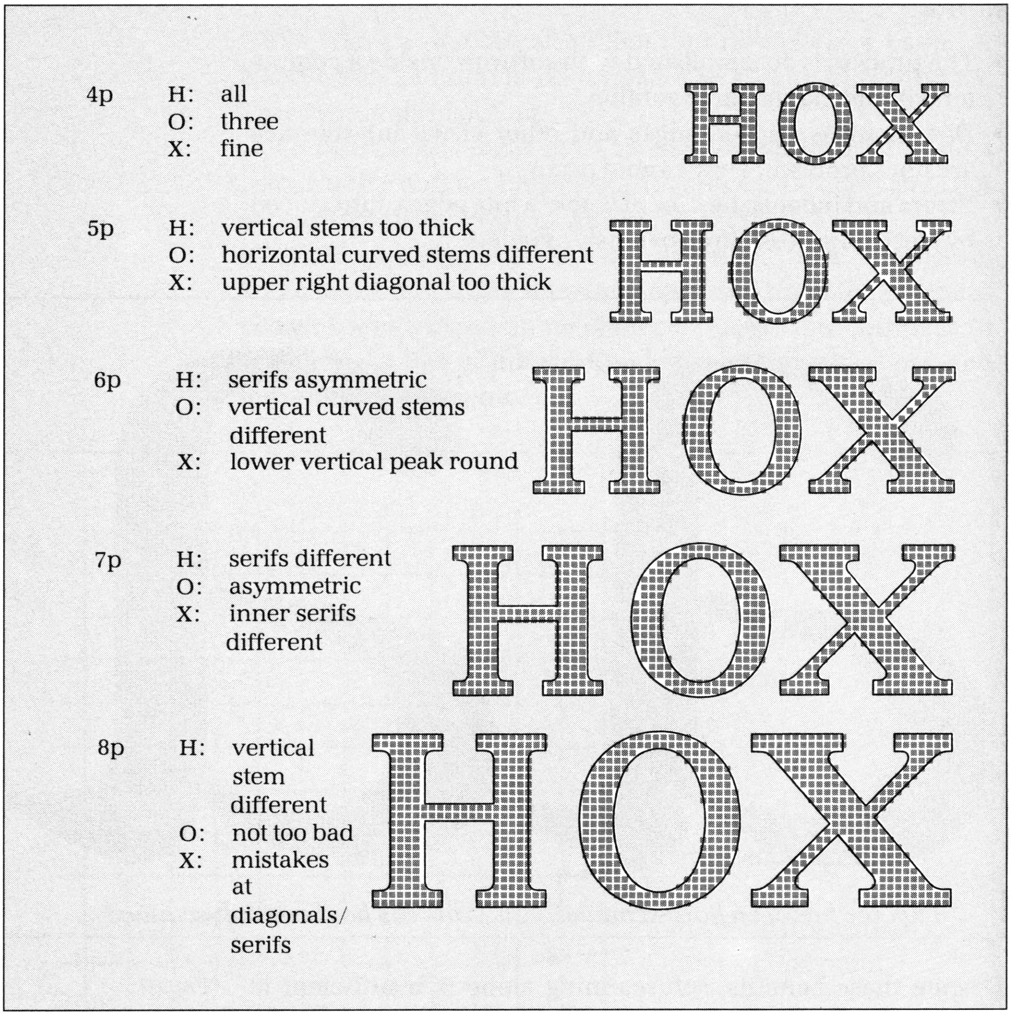


This «a» has been hard-scanned. *This «a» has been soft-scanned.*

Fig. 70
Hard- and soft-scanning:
a comparison

Despite these benefits, soft-scanning alone is insufficient to perform all font-scaling tasks necessary to produce bitmaps used in DTP. For instance, arbitrary truncation effects will sometimes be apparent in resulting bitmaps. Straight and curved stems, serifs and peaks frequently suffer severe damage, especially if they are vertical or horizontal. Diagonals are affected less from these effects.

The examples shown here illustrate the truncation effects mentioned. The outlines «H», «O» and «X» have been adjusted to be rasterized without significant truncation error at 4 point-size at 300 lines (dots) per inch. This size was chosen for optimization as small type sizes suffer the greatest damage due to this problem.



4p
H: all
O: three
X: fine

5p
H: vertical stems too thick
O: horizontal curved stems different
X: upper right diagonal too thick

6p
H: serifs asymmetric
O: vertical curved stems different
X: lower vertical peak round

7p
H: serifs different
O: asymmetric
X: inner serifs different

8p
H: vertical stem different
O: not too bad
X: mistakes at diagonals/ serifs

Fig. 71

The contours of HOX are fitted into a 4 pt raster. They undergo linear scaling and are re-rasterized to produce additional fonts in 5, 6, 7 and 8 pt sizes. Only the 4 pt raster is satisfactory. This example illustrates that one must apply intelligent font-scaling to obtain adequate results in other point sizes.

6.4.3 Scaling

Not only the 4 pointsize exhibits errors, but discrepancies can be found at all other pointsizes in virtually all characters. Errors in addition of those described already may be seen. Now, we want to introduce the term «descriptive element». Descriptive elements of the three letters shown are:

- «H» contains one horizontal and two vertical straight stems (stems and bar).
- «O» has two vertical and two horizontal curved stems (bows and arches).
- «X» has two diagonals and four peaks.
- «H» and «X» have serifs at the stem endings, consisting of two half-serifs.

These different parts, straight stem, curved stem, serif and peak, are the most important descriptive elements defining individual characters as well as the «character» of a typeface. Manipulation and rendering of these four elements can remove some disturbing irregularities in rasterization.

A recent expression, «intelligent font-scaling», refers to a process of preparing an outline prior to its rasterization or soft-scanning. The «intelligence» is included in the font description.

Early in 1977, we were involved in discussions with the Dr.-Ing. Rudolf Hell company, concerning objectionable effects of simple soft-scanning. The PA program, created within the framework of the IKARUS system, was offered as a solution to the problem. By late 1978, the PA program was functioning to Hell's satisfaction, running floating-point calculations at a rate of one character per minute.

Eight years later, Derek Kyte of Chelgraph requested a conversion of PA from floating-point to integer operation, making it run fast enough to accommodate a RIP. It was expected to take merely three work-months to complete the project, but eventually ran to two full work-years. Fall of 1987 we attained the target, a rendering rate of 20 to 30 cps on an IBM PC and 120 cps on the Chelgraph RIP. These results surpassed Adobe's operation (5 cps), Bitstream (3 cps), and Compugraphic (4 cps). Our quality at 120 cps compared favorably with Adobe's quality at 5 cps. It is believed that even greater speeds can be achieved without quality degredation.

6.4.4 Selection

Early 1987, a comparison was made of several typefaces in the IK-format outline form:

Nimbus Roman

The IK-format of Times Roman, redesigned by Berthold AG, CG Times as redesigned by Compugraphic, Tempora as redesigned by Dr.-Ing. Hell, and finally, Times Roman as redesigned by Linotype AG. We were unable to obtain data from Monotype Corp. for their version of Times, recognized as Times New Roman. A 24-point version of Linotype Times Roman was used by Adobe as master for their adaptation.

It was seen that all redesigns were relatively close to one another, discovering subtle differences in some, in some cases notable improvements overall. Refinements from all versions were selected, particularly CG Times, as the basis for our version. After application of the various preparatory steps outlined above, a new version in the BS-format was created, named Nimbus Roman.

Nimbus Sans

As a basis for this font, IK-formats of Helvetica and New Helvetica, designed by Linotype AG, Helvetica and AG Buch as redesigned by Berthold AG, Triumvirate from Compugraphic and Holsatia redesigned by Dr.-Ing. Hell. Linotype's New Helvetica was found to be the most technically perfect version, unfortunately differing substantially from the Helvetica of Adobe. Similarly to our experience with Times Roman, all versions resembled one another quite closely. We studied refinements from all versions, with particular attention to Hell's Holsatia. The necessary steps led to the BS-format version we call Nimbus Sans.

Courier

This monospaced typeface for traditional typewriter correspondence was prepared to resemble closely the IBM typewriter face of the same name.

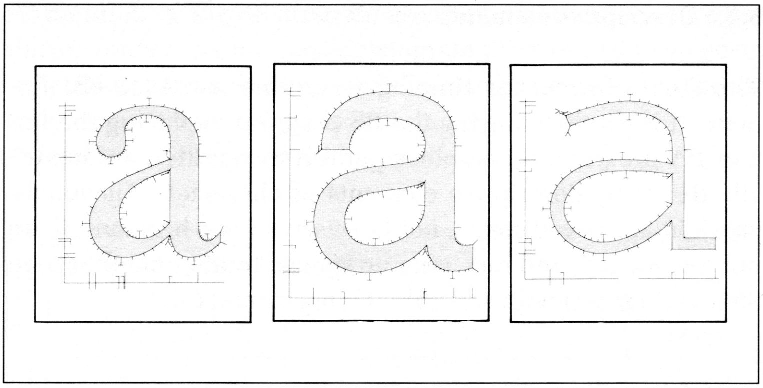


Fig. 72
Examples:
left: Nimbus Roman
middle: Nimbus Sans
right: Courier

Symbol Set

ITC's Zapf Dingbats, symbols from other typefaces, PI-fonts from our library and a number of newly designed characters were all incorporated into the URW Symbol Set. With it, we hope to set a new standard in desktop publishing.

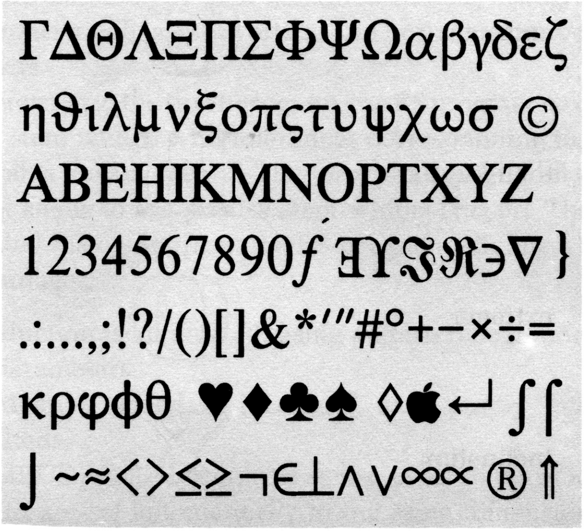

ΓΔΘΛΞΠΣΦΨΩαβγδεζ
ηϑιλμνξοπςτυψχωσ ©
ABEHIKMNOPTXYZ
1234567890ƒ ∃ϒ℘ℜ∍∇ }
.:…,;!?/()[]&*′″#°+−×÷=
κρφϕθ ♥♦♣♠ ◊ ↵ ∫⌠
⌡ ~≈〈〉≤≥¬∈⊥∧∨∞∝ ® ⇑

Fig. 73
The special symbol set contains useful characters which are not included in the basic fonts.

6.4.5 Descriptive elements

The binary format for «intelligent outline» is called «BS-format». It is produced using the BS program, replacing the former PA program. BS employs pattern recognition automatically detecting descriptive elements of characters. Diagonals, (straight or curved stems, neither vertical nor horizontal) are not recognized, and receiving no special treatment other than their association with verticals and horizontals.

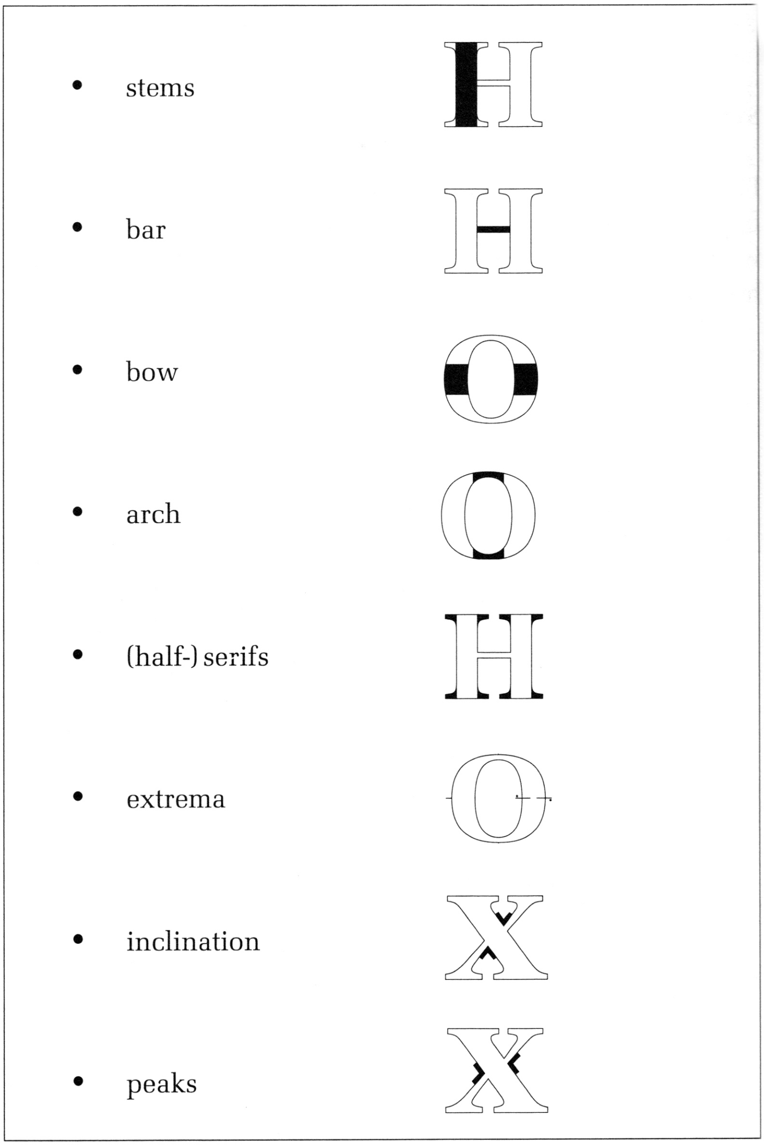
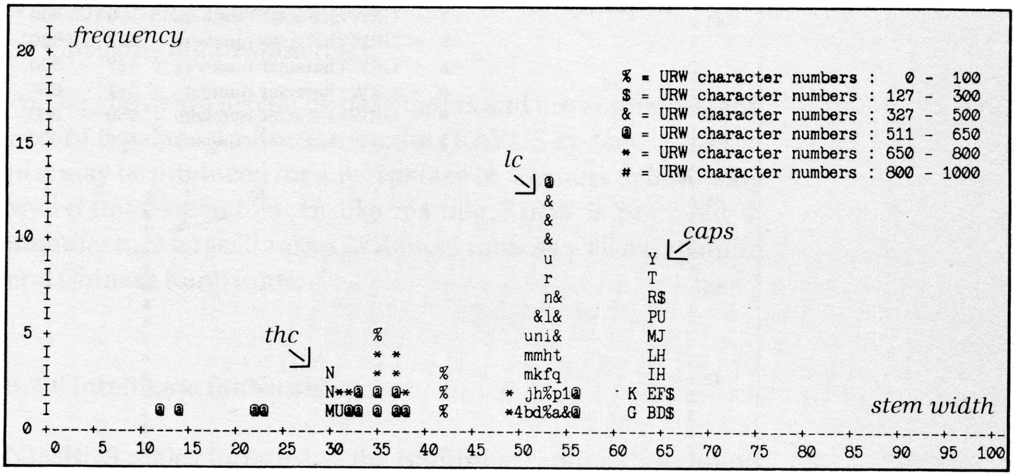

• stems
• bar
• bow
• arch
• (half-) serifs
• extrema
• inclination
• peaks

Fig. 74
The elements describing a typeface.

As mentioned, the BS program uses pattern recognition in preparing fonts to its format. A histogram is produced from vertical straight stems as an example, plotting stem widths against their frequencies at high resolution.
Twelve groups of stem widths are seen:



```
   I
20 +
   I                                                            % = URW character numbers :   0 -  100
   I                                                            $ = URW character numbers : 127 -  300
   I                                                            & = URW character numbers : 327 -  500
   I                                                            @ = URW character numbers : 511 -  650
15 +                                                            * = URW character numbers : 650 -  800
   I                                                            # = URW character numbers : 800 - 1000
   I                                                    @
   I                                                    &
   I                                                    &
10 +                                                    &
   I                                                    u          Y
   I                                                    r          T
   I                                                    n&         R$
   I                                                   &l&         PU
 5 +                                  %               uni&         MJ
   I                                  * *             mmht         LH
   I                             N    * *    %        mkfq         IH
   I                             N**@ @ @*   %      * jh%p1@     K EF$
   I           @ @       @@      MU@@ @ @@   %      *4bd%a&@     G BD$
 0 +----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+-
   0    5   10   15   20   25   30   35   40   45   50   55   60   65   70   75   80   85   90   95  100
```

Distribution of stem widths is scattered due to hand work of the craftsmen and refinements in design. Vertical straight stems are classified by their thickness. Similar histograms plot horizontal straight stems, vertical and horizontal curved stems and serifs.

Referring to the first histogram, the BS program next selects twelve stem widths with tolerances corresponding to each of the twelve mean values. The program assigns all vertical straight stems to the precise stem widths chosen. The newly generated histogram from the BS program exhibits a number of advantages:

Fig. 75
The distribution of stem widths is scattered by handwork and design refinements.

thc = thin cap stems
lc = lower case
caps= capital letters

- Production is automated, using pattern recognition driven by parameters.
- Production speed is increased over purely interactive methods.
- NIMBUS precisely controls stems, as all stems of identical width are set automatically to the same dimension in the final bitmap form.

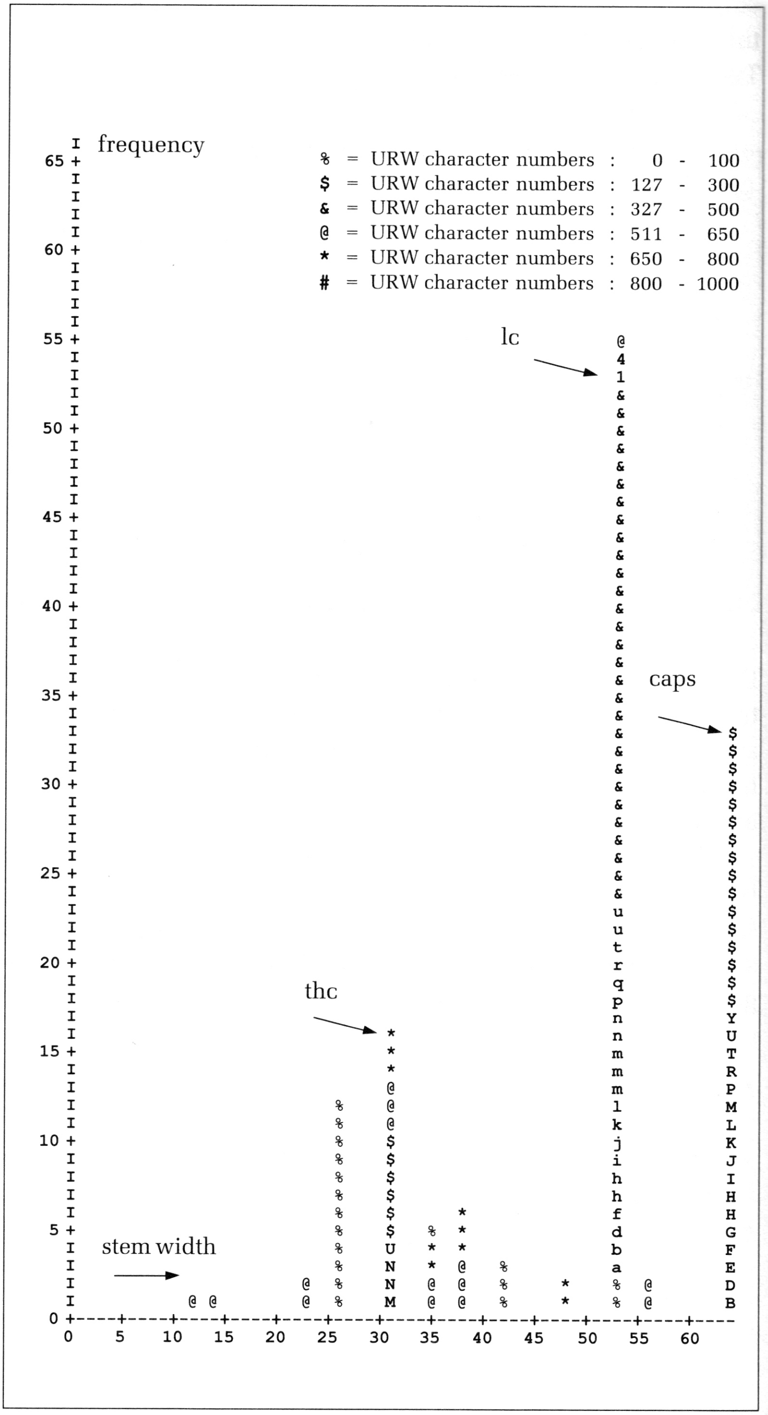


Fig. 76
When the stems have been recognized and adjusted, we have a «clean» width distribution usable for intelligent scaling in low resolution.

In actual practice, the described step is at the heart of many interrelated production techniques, including:

- Channel processing
- Insertion of half-serifs
- Unitizing
- Symmetrizing
- Rearranging extrema
- Checking digitization rules.

All the above are driven by parameters and are automated and part of typeface production on the IKARUS-system. A BS-format may be produced for any typeface in 10 hours or less using well-trained operators. In like manner, URW is prepared to manufacture large libraries of Roman fonts as well as Japanese and Chinese Kanji fonts.

6.4.6 Intelligent font-scaling

NIMBUS scales fonts using the BS-format outline description and «switches». Simple scaling and rasterization having previously been shown, the true heart of «intelligent» scaling lies in the use of switches. The BS-format contains a stem block, providing a stem with borders of:

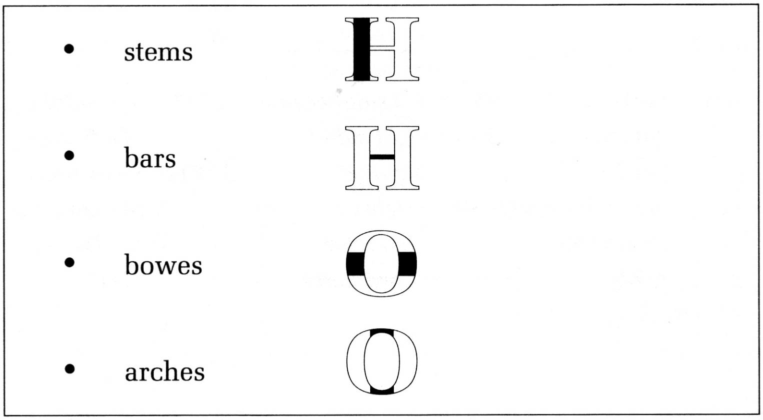


Fig. 77
The key strokes of a typeface

Every stem has two borders on either side. Normally, one border position is coded absolutely and the second relative to it. Consider the two vertical curved stems of a capital «O» and what occurs during the process of stem adjustment.

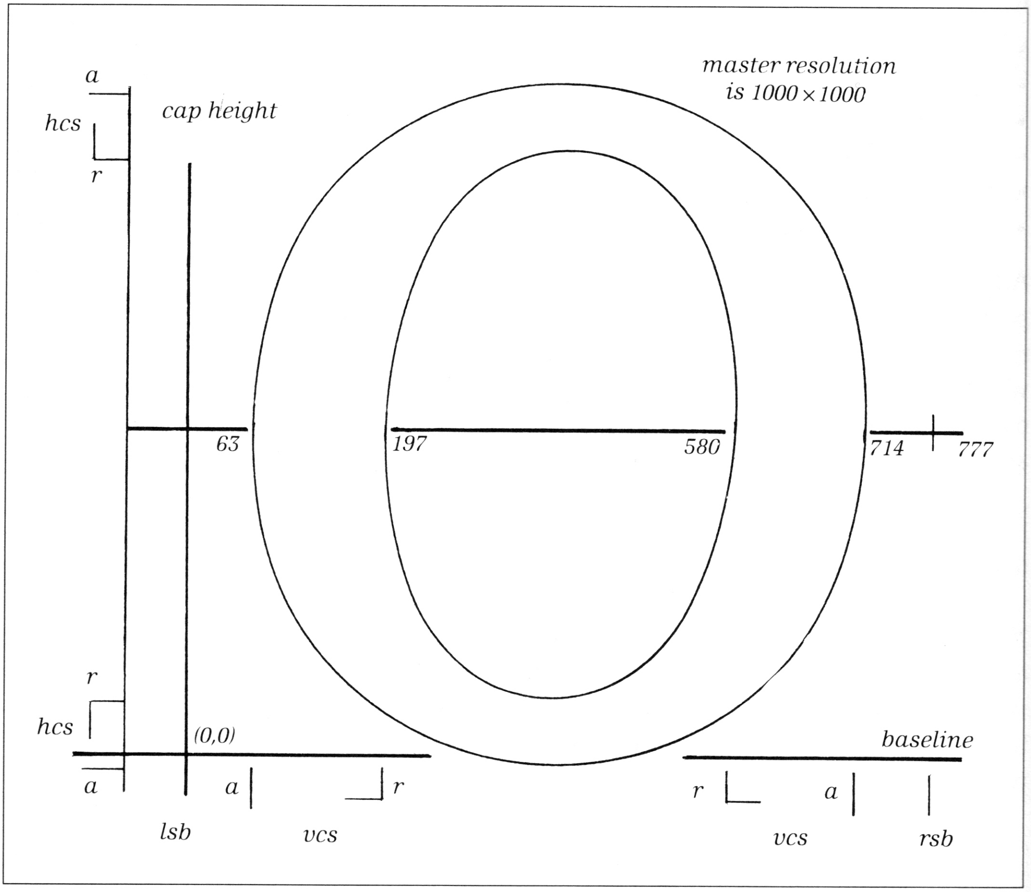

a
hcs
r
cap height
master resolution is 1000 × 1000
63
197
580
714
777
r
hcs
(0,0)
baseline
a
a
r
r
a
lsb
vcs
vcs
rsb

Fig. 78
Here the curves of the «O» are checked.

hcs = horizontal curved stem border
vcs = vertical curved stem border
r = relative
a = absolute
lsb = left side bearing
rsb = right side bearing
(0,0) = origin of character coordinates
63, 197 = x-positions of left stem
580,714 = x-positions of right stem
777 = total width of «O»

Stem adjustment

Given a master resolution of 1,000 × 1,000 pixels per em, the task of creating from it a 10 point image to be applied to a 300 dpi laser printer, NIMBUS divides coordinates of the outline by a factor of 25 to obtain an em square of 40 × 40 pixels.

Therefore, after rounding-off (pure rasterization) we get:

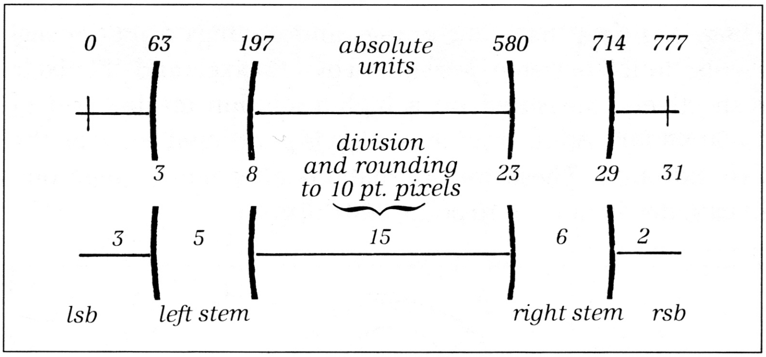

0
63
197
absolute units
580
714
777
division and rounding to 10 pt. pixels
3
8
23
29
31
3
5
15
6
2
lsb
left stem
right stem
rsb

Fig. 79

One sees:

- unequal bearings (3 pixels and 2 pixels)
- unequal stem widths (5 pixels and 6 pixels)

That is an unpleasant result. Therefore, first, NIMBUS detects all stem borders using relative distances. These absolute positions provide equal bearings and equal stem widths, seen here:

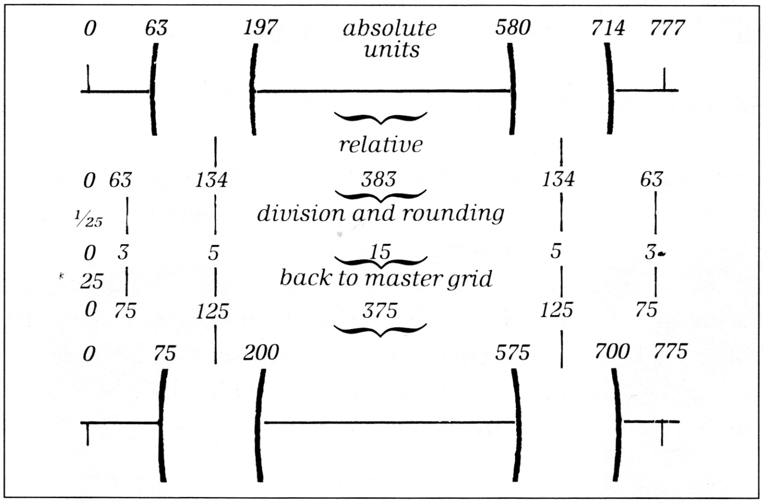

0
63
197
absolute units
580
714
777
relative
0 63
134
383
134
63
1/25
division and rounding
0 3
5
15
5
3
25
back to master grid
0 75
125
375
125
75
0
75
200
575
700
775

Fig. 80

- bearings of 3 pixels each,
- stem widths of 5 pixels each.

Curve switches are used to rearrange curve outlines. The stem border positioned furthest left is moved from position 63 to position 75. We have switches for:

- curve X-extreme and
- curve Y-extreme.

These switches mark beginnings and endings of curve segments: Both X-extremes are shifted by +12 pixels and - 14 pixels respectively, measured on a high-resolution master grid of 1,000 × 1,000. A linear interpolation is performed between the two extremes. These transformations alter letter shape outlines suited better to a 10 point, 300 dpi grid.

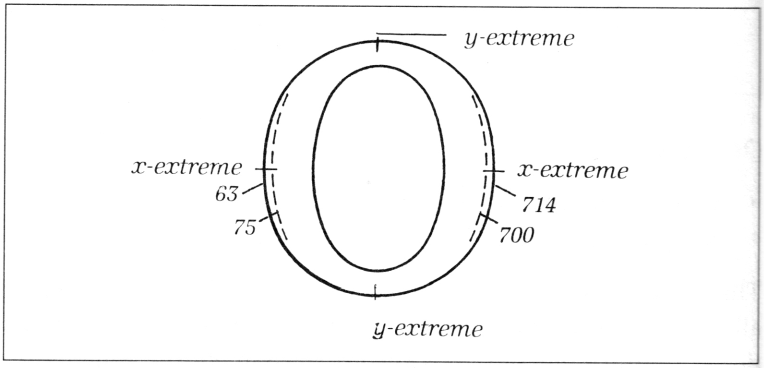


Fig. 81
Grid fitting with intelligence

Switches for curve starts and ends allow curves to be adjusted harmoniously with stems, corners and other curves.

Serif adjustment

Uppercase (cap) «O» is a simple challenge for NIMBUS. With it, we can illustrate how it works. Serif adjustment is still easier – usually only half-serifs are adjusted.

With help from serif start and end switches, half-serifs are removed and stems adjusted without them. Whenever a stem border is shifted, the corresponding half-serif follows. Only one half-serif is inserted during digitizing. Other halves are calculated by electronic mirroring.

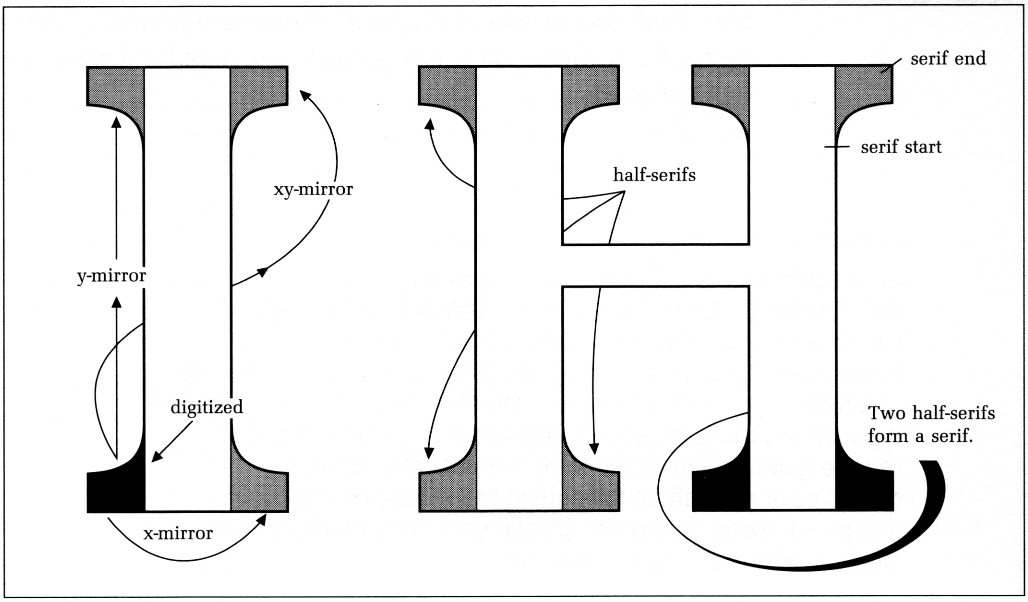
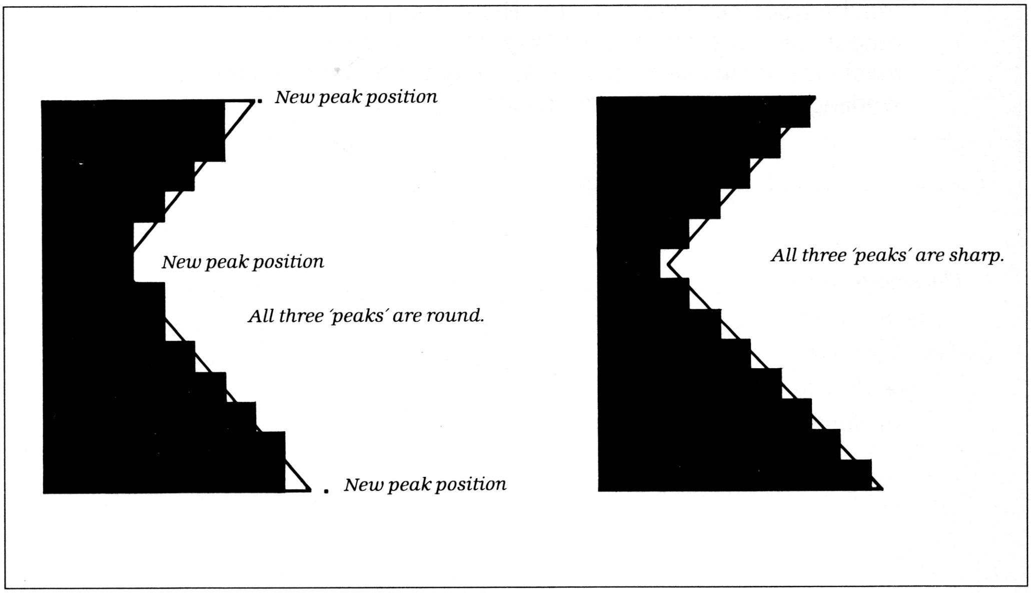

serif end
serif start
xy-mirror
half-serifs
y-mirror
digitized
Two half-serifs form a serif.
x-mirror

Fig. 82
Serifs are handled as attachments to stems.

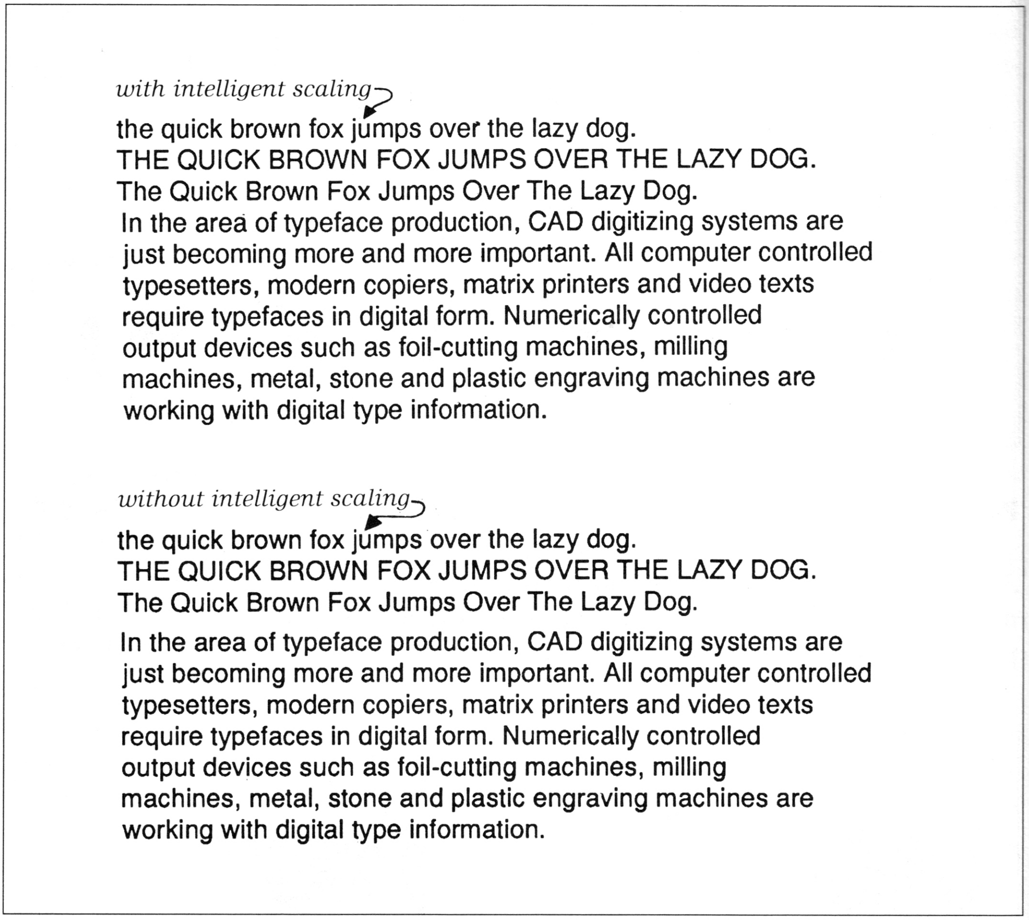
Peak adjustment

A sharp peak intended to remain sharp can accidentally terminate in two pixels instead of a single one. A small shift in peak position brings surprisingly satisfactory results.

Fig. 83
A slight shift of the resolution mark delivers better results.


New peak position
New peak position
All three 'peaks' are round.
New peak position
All three 'peaks' are sharp.

NIMBUS adjusts each peak automatically to its optimal position. Peaks occur where diagonal strokes are joined by other stems. An artificial shape has been chosen here for demonstration purposes.



with intelligent scaling

the quick brown fox jumps over the lazy dog.
THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG.
The Quick Brown Fox Jumps Over The Lazy Dog.

In the area of typeface production, CAD digitizing systems are just becoming more and more important. All computer controlled typesetters, modern copiers, matrix printers and video texts require typefaces in digital form. Numerically controlled output devices such as foil-cutting machines, milling machines, metal, stone and plastic engraving machines are working with digital type information.

without intelligent scaling

the quick brown fox jumps over the lazy dog.
THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG.
The Quick Brown Fox Jumps Over The Lazy Dog.

In the area of typeface production, CAD digitizing systems are just becoming more and more important. All computer controlled typesetters, modern copiers, matrix printers and video texts require typefaces in digital form. Numerically controlled output devices such as foil-cutting machines, milling machines, metal, stone and plastic engraving machines are working with digital type information.

Fig. 84
The significance of intelligent font-scaling-results «with» and «without».

6.4.7 The resolution funnel

NIMBUS is capable of generating all pointsizes from 4 to 270 point for use on impact and non-impact printers writing at 240, 300 and up to 1,200 dpi. NIMBUS also generates run length codes for high resolution typesetters.

At low resolution, it drops to a somewhat lower level of performance. Higher than 18 × 18 pixels (less than 4-1/2 point at 300 dpi) NIMBUS finds an optimum solution. Sizes smaller than 18 × 18 pixels, considered as difficult, are normally used only for «What You See Is What You Get» (WYSIWYG) editing screens. Even at these sizes, NIMBUS reasonably represents the texture of a typeface.

The highest possible quality screen display should be used in day-long text composition applications. To facilitate this kind of refinement, NIMBUS is accompanied by a series of hand-edited bitmaps.

The value of intelligent scaling.

Assuming a proportional text typeface contains 300 characters, and that a 16 × 16 pixel bitmap could be stored in 32 bytes, an upper limit storage capacity can be calculated to be 9.6 Kbytes per screen font. Storing 142 proportional fonts requires about 1.4 Mbytes, little more than half a 16 Megabit chip.

As this satisfies just one aspect ratio, we also provide the following others:

Device	Resolution (lpi)	Aspect ratio
Macintosh	72×72	1:1
CGA	96×48	2:1
EGA	96×72	4:3
VGA	96×96	1:1

NIMBUS is supplied with four sets of hand-edited bitmap screen fonts at 6 to 14 points in size. NIMBUS itself will successfully generate output for sizes larger than this.

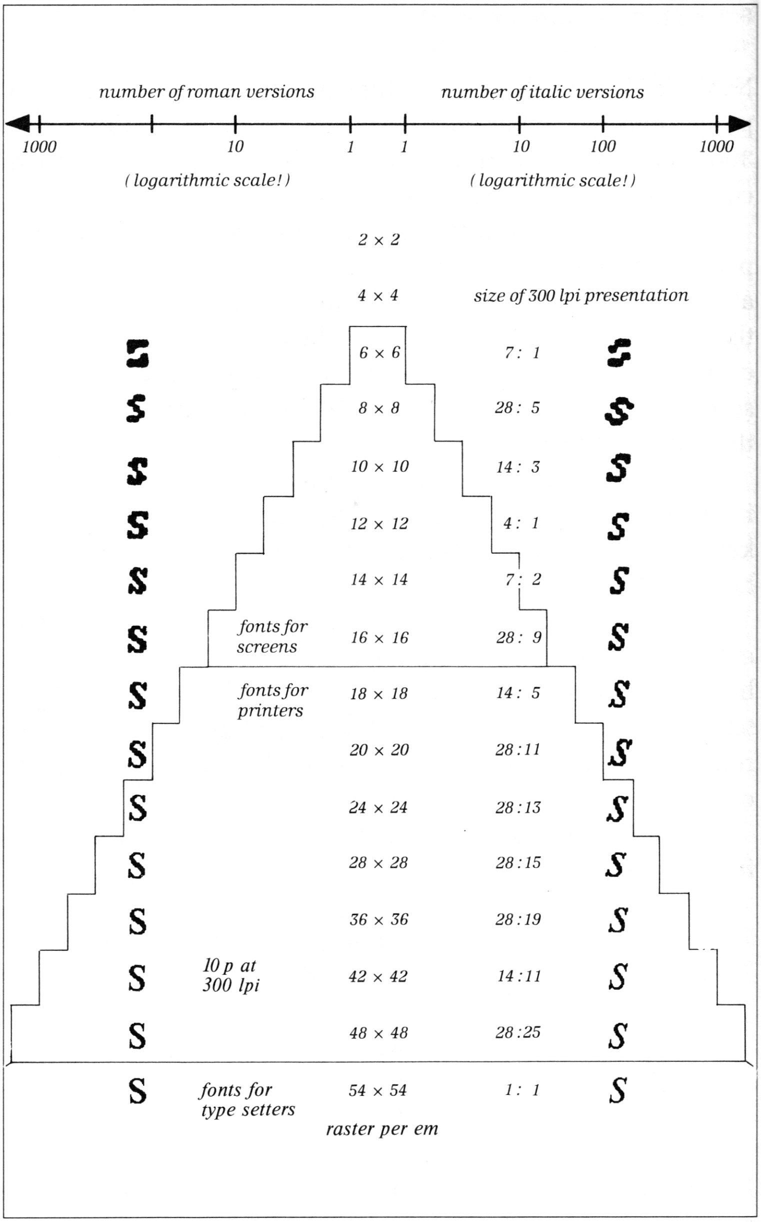

number of roman versions
number of italic versions
1000
10
1
1
10
100
1000
(logarithmic scale!)
(logarithmic scale!)
2 × 2
4 × 4
size of 300 lpi presentation
6 × 6
7 : 1
8 × 8
28 : 5
10 × 10
14 : 3
12 × 12
4 : 1
14 × 14
7 : 2
fonts for screens
16 × 16
28 : 9
fonts for printers
18 × 18
14 : 5
20 × 20
28 : 11
24 × 24
28 : 13
28 × 28
28 : 15
36 × 36
28 : 19
10 p at 300 lpi
42 × 42
14 : 11
48 × 48
28 : 25
fonts for type setters
54 × 54
1 : 1
raster per em

Fig. 85
With progessively coarser resolutions, fewer versions of each font can be represented. At a raster of 6 x 6 for the em, we can essentially display only one version, the «Helltimes» or «screenfont».

Naturally, in each case we can also add a monospace font.

Possible number of fonts (for the em raster)

2	*(6p)*
4	*(7p)*
6	*(8p)*
10	*(9p)*
10	*(10p)*
20	*(11p)*
20	*(12p)*
30	*(14p)*
40	*(16p)*

142 fonts

Chapter 7

The preparation of rasters

In previous chapters it was explained that open vector formats, run length codes or bitmaps are no more than special descriptions of that which is understood as digital type, essentially, letters presented in a particular raster form.

This statement description also applies ultimately to any format using curved lines to describe digitization. In our discussion here, their resolution is so fine (the DI-format is 15,000 x 15,000 units per em) that we are no longer so concerned with the raster itself, but only details such as cap height or thickness, or in the case of machine cutting, accuracy of analog reproduction (see point 3.3). But, back to raster forms.

Em rasters sizes between 10 × 8 and 1,600 × 1,600

At URW, generation of scanline rasters is referred to as preparation of SC-formats. The raster size per em can be chosen arbitrarily in a range from 10 × 8 (the lowest) to 1,600 × 1,600 (very fine resolution). Video display units or dot matrix printers work successfully with 2 or 3 lines/mm, while better typesetters use 100 lines/mm. In the latter instance, a raster of 1,600 × 1,600 units per em would translate to a body size of 16 mm = 42.7 pt = «display of headlines in text». It is possible to create rasters using disimilar separations between X and Y scanlines. Two steps are used by URW to prepare the SC-format. The first is automatic soft-scanning, the second, any possible residual adjustment due to interactive earlier correction on a graphic display. Use of the phrase «possible residual adjustment» is deliberate, as most soft-scanned letters, when done properly, do not require any further correction.

Some applications have completely disregarded even normally unwanted scanning artifacts. Rastered type has, in fact, been marketed in which the direct results of hard-scanning normal artwork – marks, contaminations, flaws, etc. caused by electronic noise – were purposely left intact in order to get to market quickly.

7.1 Soft-scanning

URW considers soft-scanning to be a fairly well established procedure, developed over years, and proven to be a correct approach.

First the contour line is moved.

It was necessary to recognize that the precise DI-format must be viewed as thin lines superimposed upon a usually more coarse raster. Overall appearance of the rasterized image is scarcely affected with discrepancies, caused by fine grid adjustments made within the DI lines. Corresponding corrections or adjustments were programmed in to automatically achieve the effects described next.

Channel processing

The «Channeling» program places nearly vertical or horizontal straight lines in precise vertical and horizontal orientation, resulting in smooth reproduction quality without raster skips or «jaggies». The same program additionally deals with these situations:

- All curves with only slight curvature, such as flutes in a serif, are not processed by channeling.
- Diagonals or even slightly sloped straights are likewise not affected by the process.

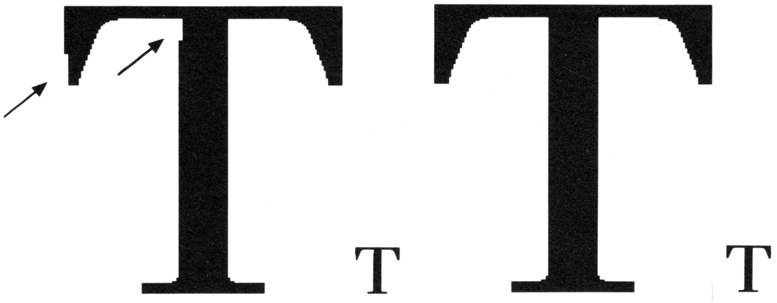


Fig.86 Automatic channel processing ensures sharp, straight lines.

Grid fitting of straight stems and crossbars

The «PA» program for «grid fitting» organizes stems of approximately equal thickness, which through the rastering process would emerge slightly non-uniform, and fits them into identical raster lines. This fitting should not, as far as is possible, change the appearance of the original, nor should a change of weight occur.
The program takes the following matters into account:

- Straight stems with ductus changes, as with «Optima», are not affected.
- Fitting is accommodated for raster sizes up to approximately 400 × 400 units per em. Finer rastering has eliminated the need for such detailed treatment.
- Diagonals are not treated at all.

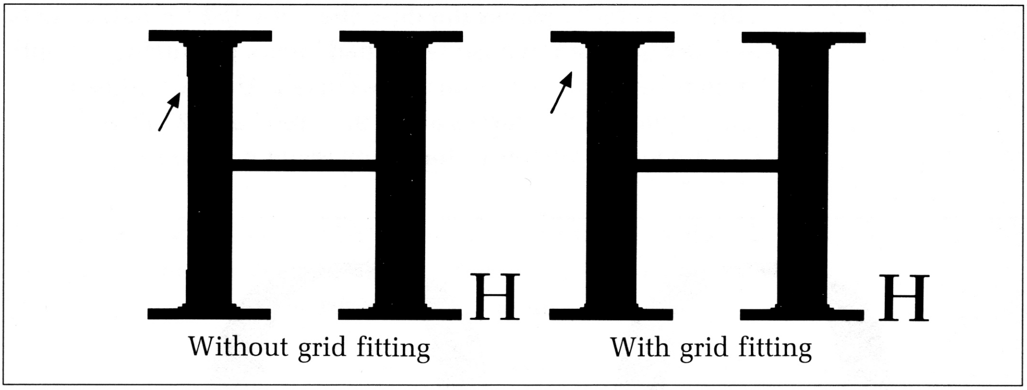
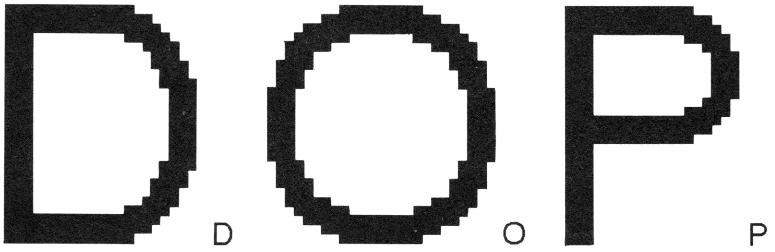


Fig. 87
Grid fitting for stems improves the serifs.

Thickness of horizontal and vertical curved strokes

Curved strokes of uniform thickness in a typeface, altered nonuniform by rasterization, can be specifically refitted to the same number of raster lines in horizontal and vertical segments.

Curved paths can also be automatically adjusted to a prescribed raster. Particularly, light weight type in smaller sizes would display large numbers of inconsistencies if such adjustments were not made.

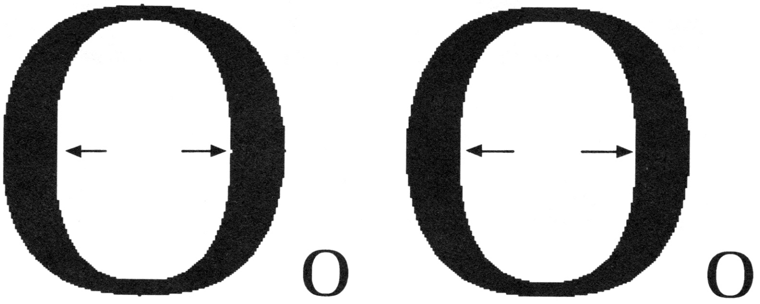


With grid fitting of straights and curves

Fig. 88
Low-resolution rasters require line fitting.

Curves at 0°, 90°, 180° and 270°

This adjustment places the described line (DI-format) exactly midway between two «soft-scanned lines», producing the optimum raster representation of the curves. After the adjustment, the extrema of the curves at 0°, 90°, 180° and 270° no longer contain either too few or too many raster points.



Without curve fitting

With curve fitting

Fig. 89
Curve fitting is important.

The following sketches attempt to illustrate this effect as clearly as possible:

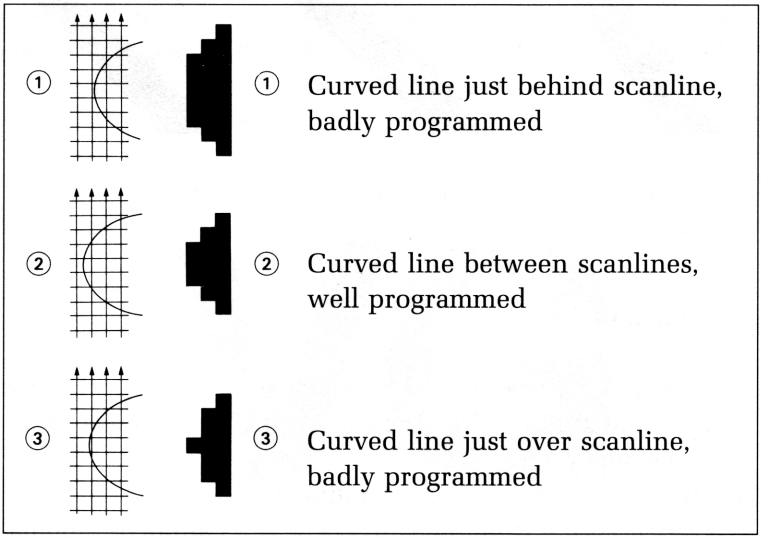

①
① Curved line just behind scanline, badly programmed
②
② Curved line between scanlines, well programmed
③
③ Curved line just over scanline, badly programmed

Fig. 90
Optimum placement of curves on raster

Curves at 45°, 135°, 225° and 315°

Curved segments at 45°, 135°, 225° and 315° are treated similarly in a situation where a distinct step structure may occur resulting in questionable reproduction quality.

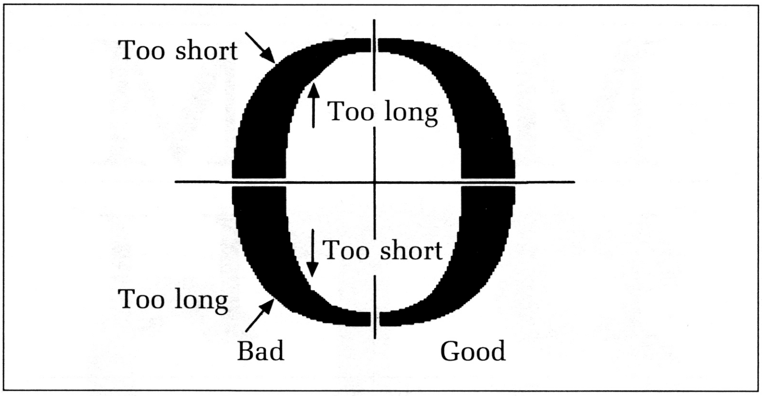

Too short
Too long
Too short
Too long
Bad
Good

Fig. 91
Optimum fitting is also possible under «45°».

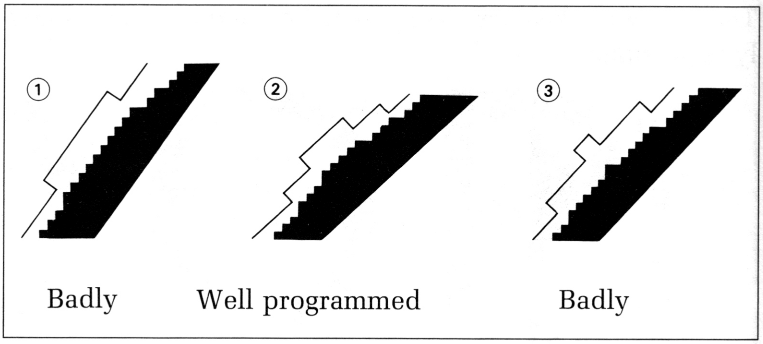

①
②
③
Badly
Well programmed
Badly

Fig. 92
Details from curves under «45°»

Treatment of serifs

- Joins of the curves from horizontal serifs to vertical stems are made uniform. Lengths and heights of serifs are equalized where required.

Always to be taken into consideration are:

- Vertical serifs, as found in an «E» or «F», cannot be automatically adjusted.
- Serifs whose upper or lower edges are not parallel to the base line cannot be adjusted.
- Serifs occurring on diagonal stems, likewise, cannot be adjusted.

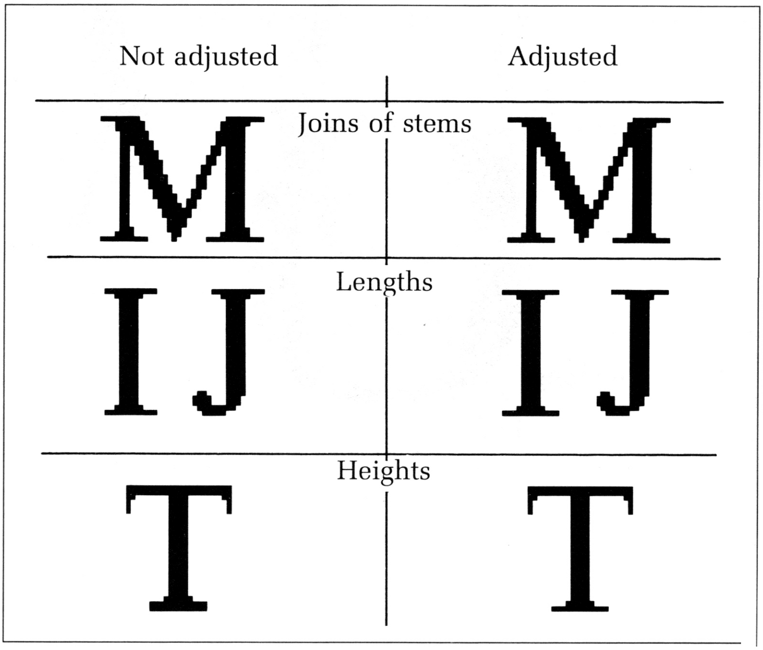

Not adjusted
Adjusted
Joins of stems
M
M
Lengths
I J
I J
Heights
T
T

Fig. 93
Further examples of fitting

Treatment of individual raster points

Raster dots or points entirely isolated or with but a single diagonal connection to a neighboring dot, may be automatically deleted. Those points, black or white, with either vertical or horizontal contiguous connection to another point will not be automatically deleted.

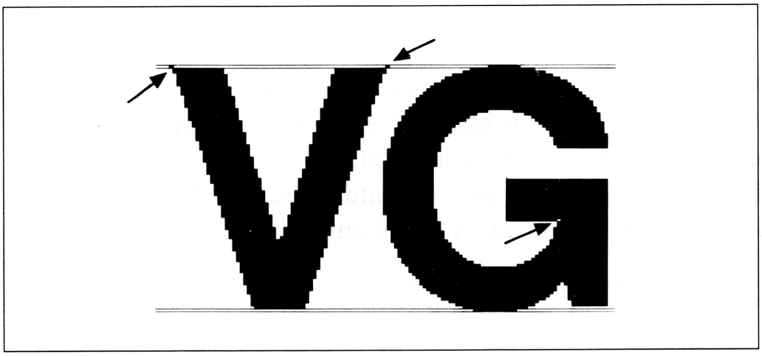


Fig. 94
Treatment of apex points

Height adjustment of the characters of a font

The standard characters of a font (lowercase, caps or uppercase, numerals) are adjusted using raster steps matching prescribed auxiliary lines. Such lines lie on a calculated raster line of the respective scanline step. Determining factors are the character's descender, baseline, x-height, cap height, lowercase ascender height and overhangs of curves.

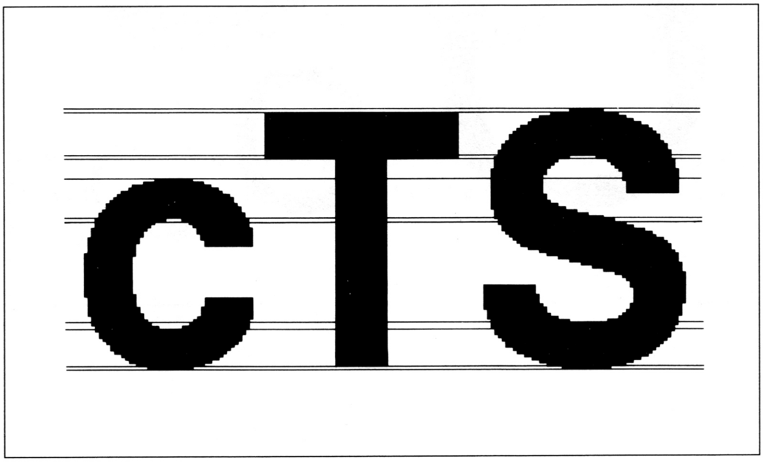


Fig. 95
Adherence to auxiliar lines

Keep in mind, generally, these suggestions:

- The more exotic the face, the less adjustment should be made.
- Special characters and punctuation marks occasionally extend beyond the normal boundaries, and, for that reason, should not be adjusted.

Identity of various pointsizes

Differing pointsize of a typeface may not reproduce identically if the raster size is changed as there are only full, not half pixels. The table below lists guideline parameters for cap height (H), vertical stems (X) and crossbars (Y) measured in scanlines, for rastering in 32, 16 and 8 point examples:

	32pt	16pt	8pt
H	84	42	21
X	11	6	3
Y	10	5	3

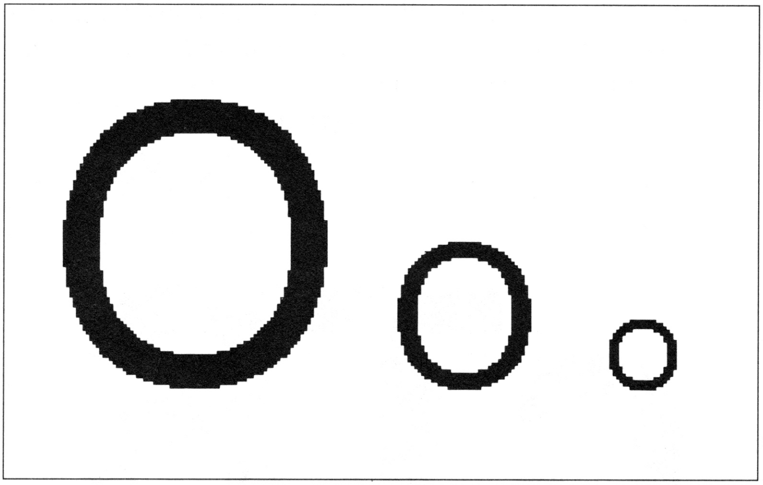


Fig. 96
Rasters of the letter «O» in 32pt, 16pt, and 8pt (rasterized from top to bottom)

Because not all measures of 32 point letters are even, 16 pt. size cannot be produced by simply halving. The 8 pt. size, also, must be changed as halving uneven numbers of scanlines, up or down, will not precisely match the 32 or 16 pt versions. In actuality, the 8 pt. size will prove to be bolder in weight.

Yet, pointsizes can be matched when working «from bottom to top». The parameter for the smallest pointsize is calculated and is increased by multiplication. For this example, the parameter list would be as follows:

	32pt	16pt	8pt
H	84	42	21
X	12	6	3
Y	12	6	3

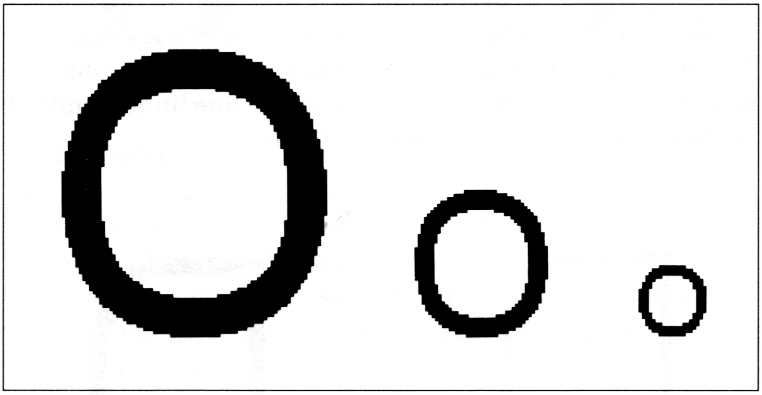


Fig. 97
Rasters of the letter «O» in 32pt, 16pt, and 8pt (rasterized from bottom to top)

The result now displays greater deviation from the original. Heights, stem thicknesses, etc., however, are consistent for all three pointsizes.

Disputable cases

The following examples demonstrate how a non-adjusted letter of 1.5 pixel stroke width is adjusted to two different stem thicknesses. The first uses one scanline, the other uses two scanlines giving corresponding stem parameters to the grid fitting procedure.

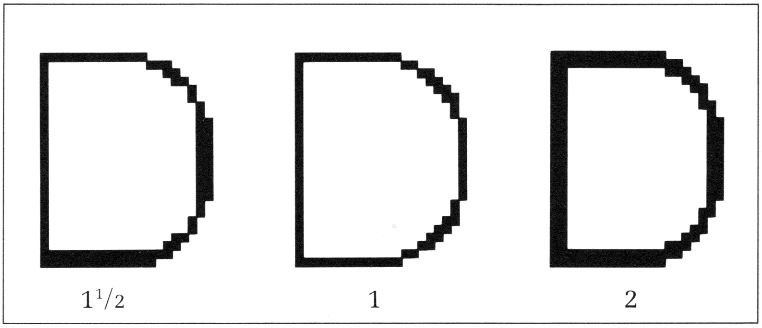


Fig. 98
What can result from a stroke width of 1.5 pixels.

1¹/2 = without fitting
1 = with grid fitting of stroke width to one pixel
2 = with grid fitting of stroke width to two pixels.

As all three results above are proven unsatisfactory, we see that soft-scanning alone is no solution. The contouring program must be used before thinning to one line (inner outline), or thickening to two lines (outer outline).

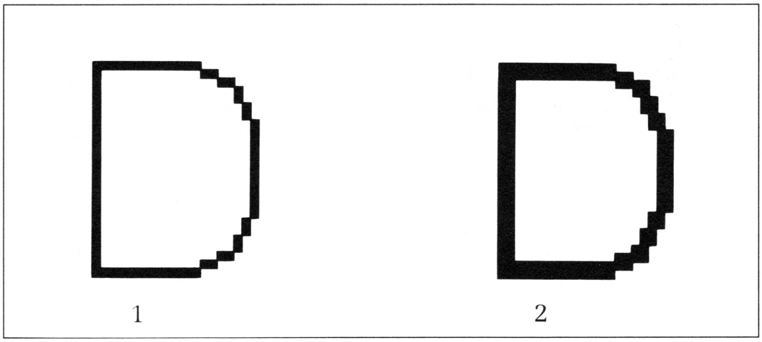


Fig. 99
Two possible solutions for a stroke widtsh of 1.5 pixels

1 = with grid fitting of stroke width to one pixel and thinning of curves by automatic inlining.
2 = with grid fitting of stroke width to two pixels and thickening of curves by automatic outlining.

7.2 Manual corrections

In examples to follow problems of utilizing automatic correction are brought to the reader's attention in order to stress the importance of delivering as perfect type reproduction as practicable.

7.2.1 Corrections on stems

Straight stems

Letters comprised originally of slightly curved stems may have those stems straightened when a raster is too coarse. The resulting stems often cannot be automatically adjusted and must be corrected manually.

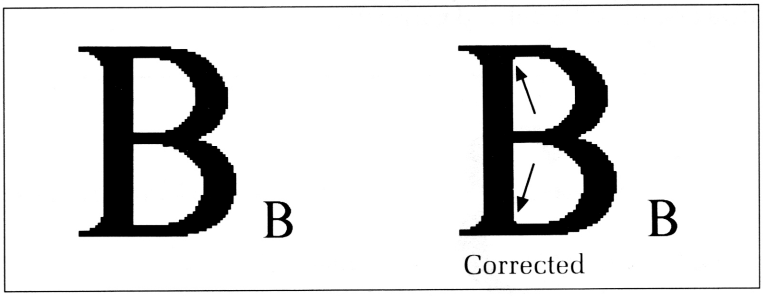


Fig. 100
Treatment of «rounded» corners

Stroke thicknesses

Overly coarse rasters can lead to loss of fine differentiations in stroke thickness. The lighter or heavier a typeface – compared to a regular weight – and the more coarse the display, the greater difficulty it is deciding for or against a scanline. The loss occurs in horizontal, vertical, diagonal as well as curved strokes.

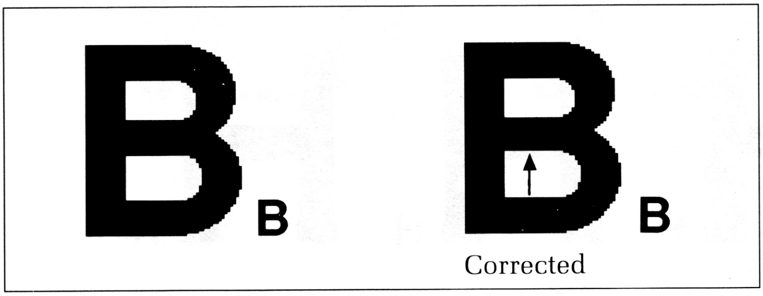


Fig. 101
Treatment of three cross stems

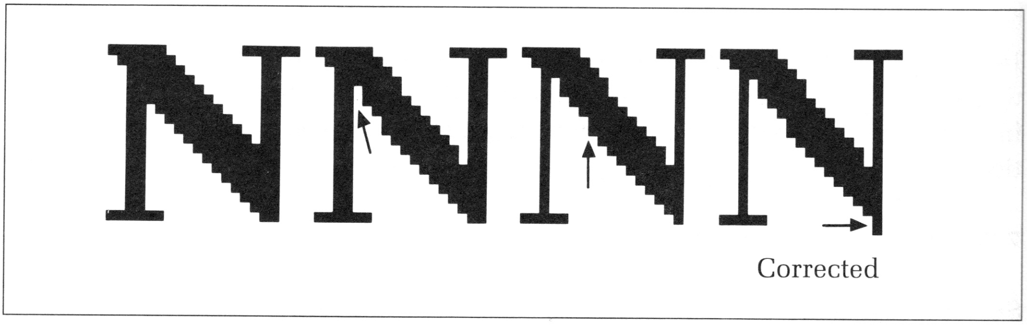


Fig. 102
Possible solutions for the N

Flat curves become straight

A very coarse raster turns flat curves into straights. In the example below, only a single point is left of the original curvature and must be deleted manually.

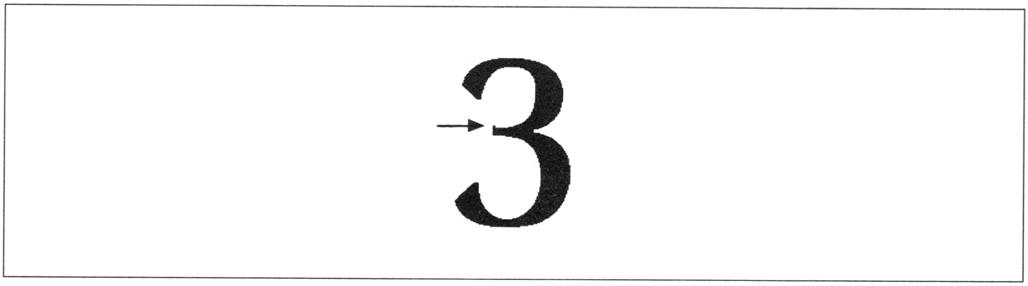
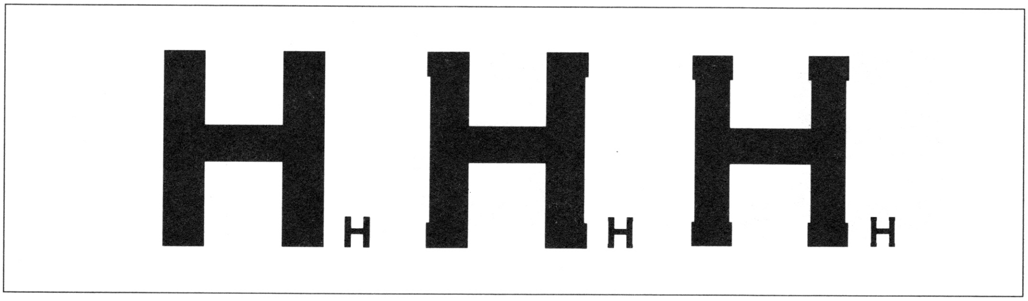


Fig. 103
Straight or round?

Stems with changes in ductus

Fig. 104
The «optima» problem

Stems only slightly curved inward or outward can be neither straightened nor adjusted. To keep them uniform, they must be manually adjusted point for point.



Scanline skips

Scanline skips (jaggies) appearing in curves, which are detrimental or detract from a visual appearance, must be manually corrected.

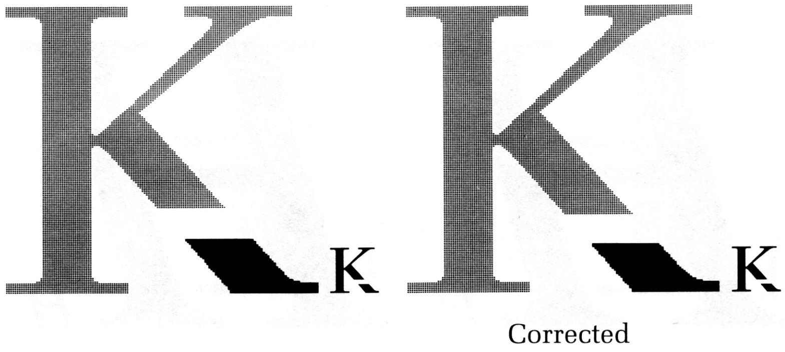
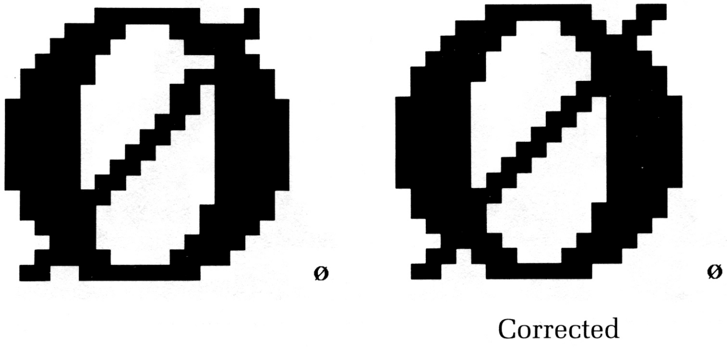


Fig. 105
Serifs on diagonal

Discontinuous curves

At the intersection of curved strokes and other line segments, no possibility exists for automatic adjustment. Again, manual intervention is required.

Fig. 106
Discontinuous diagonals

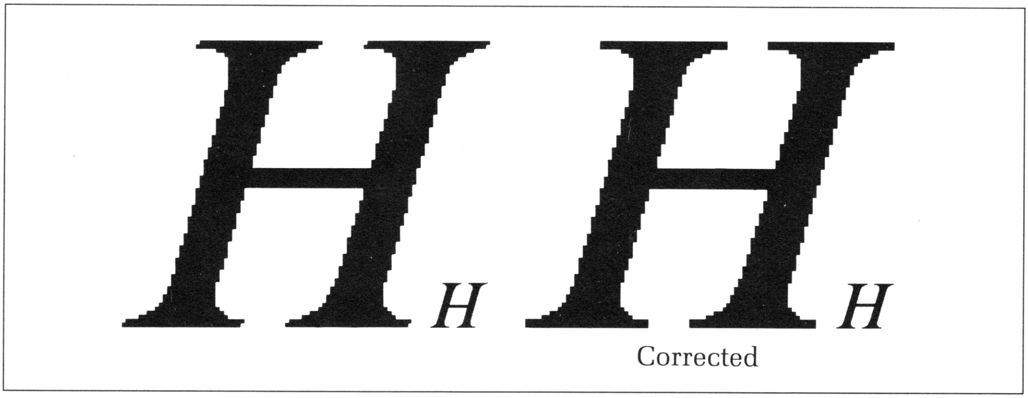


7.2.2 Correction of serifs

Serifs of italic type

Fig. 107
Italic Serifs

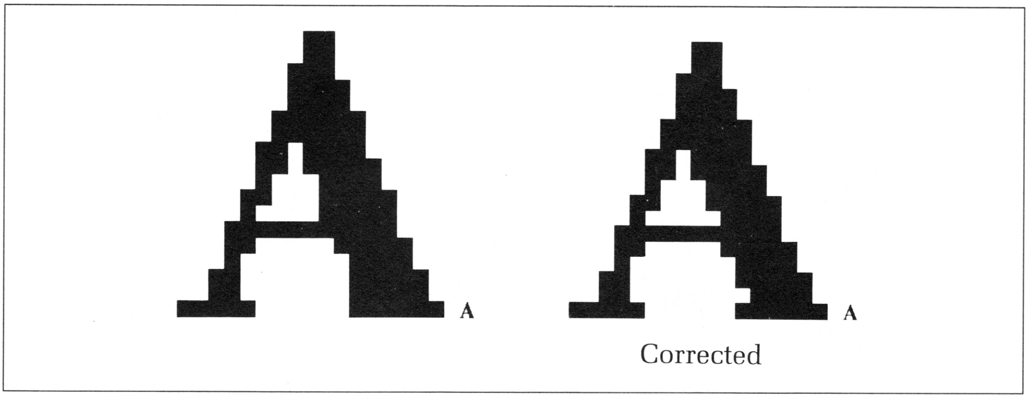
Ends of serifs become problematic if a coarse raster is used. In the example shown, serif ends have an inclination of 0°. Deleting a single point creates a 45° slope.



Serifs on diagonal stems

Fig. 108
Coarse raster effects peaks and serifs.

Serifs terminating a diagonal stem cannot be automatically adjusted. If using a coarse raster, corrections are necessary in order to achieve a smooth transition from stem to serif.



Flutes on serifs

As with diagonal serifs, flutes on serifs cannot be automatically treated. Using a coarse raster, it must be decided how many scanlines need to be left out and uniform flutes usually must be generated manually.

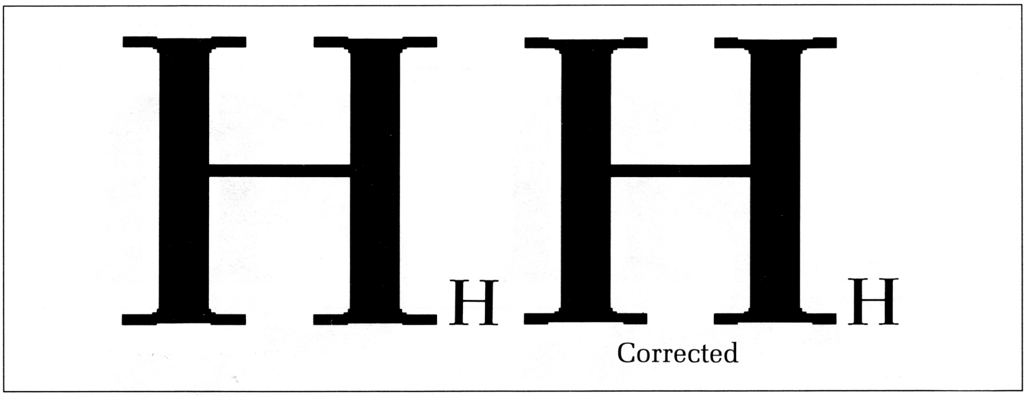
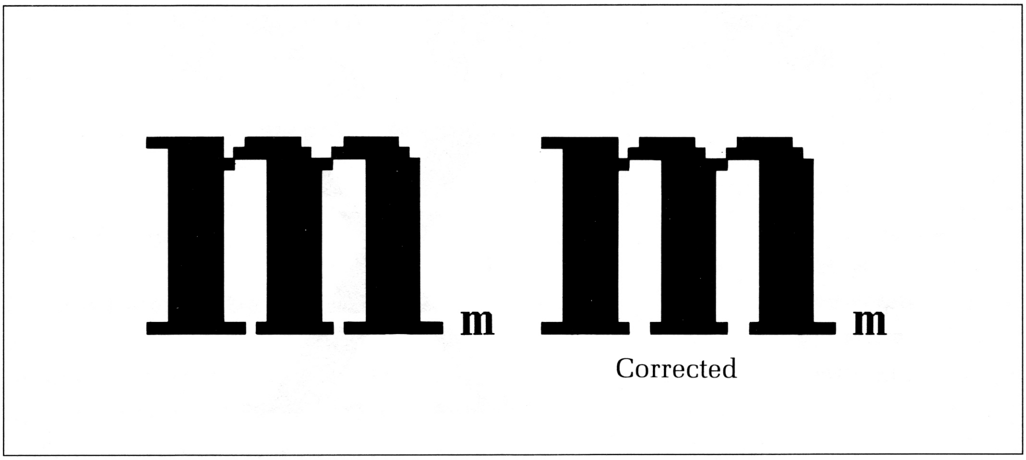


Fig. 109
Serifs with flutes

Serif lengths

Using a still more coarse raster, serifs on the letters «m», «n», «w», «v», «y», etc., can easily «intergrow» or collide with each other. In some cases straight stems (the letter m) can be dealt with automatically, but diagonals always have to be corrected manually.

Fig. 110
Shorten inner serifs if necessary



7.2.3 Corrections on segments of letters

Certain unique letter segments (apart from horizontal serifs) only retain their uniform appearance with interactive or manual correction.

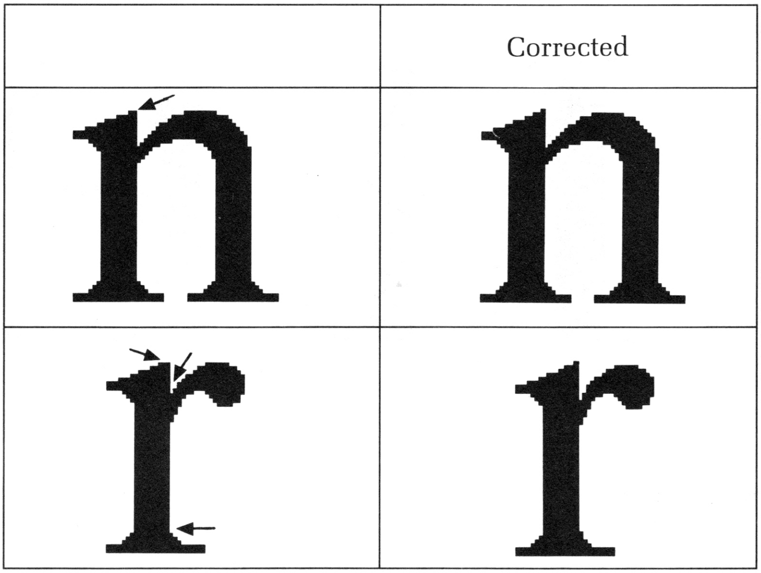


Fig. 111
Interactive correction is sometimes unavoidable.

7.2.4 Intersection of diagonals

Rastering at intersections of two diagonals frequently creates a random appearing shape at that location. Such intersection must be checked and if necessary, corrected manually.

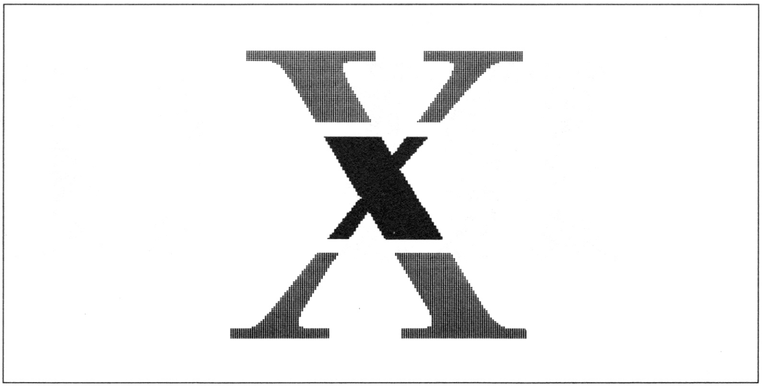
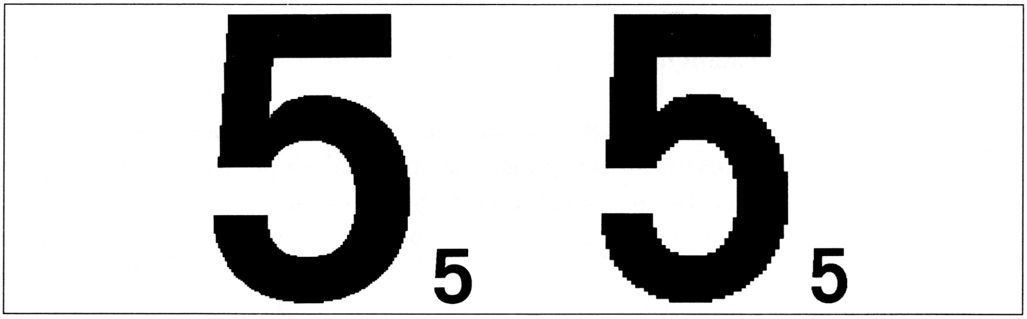


Fig. 112
Resolution marks are preserved.

7.2.5 Stems with slight slopes

When a raster is overly coarse, problems arise if slightly sloping stems are encountered. The more coarse the raster, the fewer available steps are available to describe an inclination. For instance, a nearly vertical stem in the figure 5 will certainly disappear when a raster reaches a significant degree of coarseness.

Fig. 113
Some decisions are drastic - but correct.



Special characters, such as «*», the horizontal stroke may become simply rectangular as a result of coarse rastering.

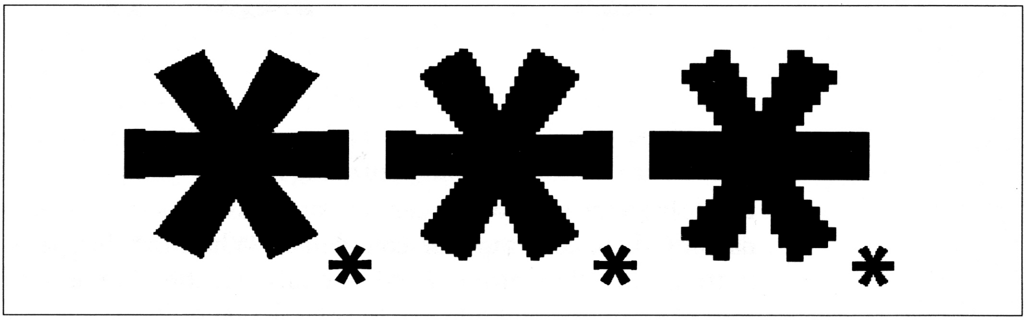


Fig. 114
Coarse rasters are a real problem for the appearance of type.

7.2.6 Ink traps

The optimum ink trap

Acutely angled joining of strokes in letters, are often cut relatively deeply for improved repro or printing purposes. This cut is achieved on raster letters by removing points. Adjustment of, say the letter M, is possible only with manual correction.

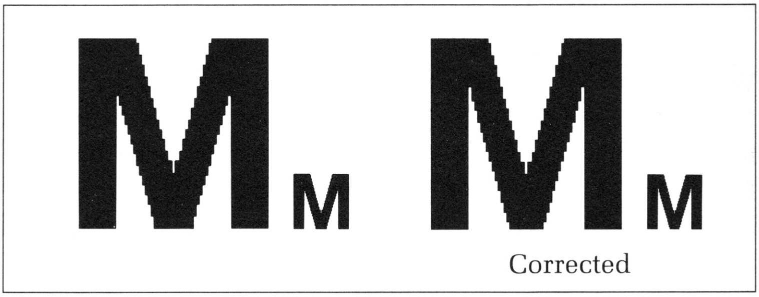


Fig. 115
Ink traps are stressed.

Changing ink traps

Coarse rastering sometimes creates distortions or exaggerations in stroke joins. The original slope will sometimes be changed by this in letters such as «r» and «b».

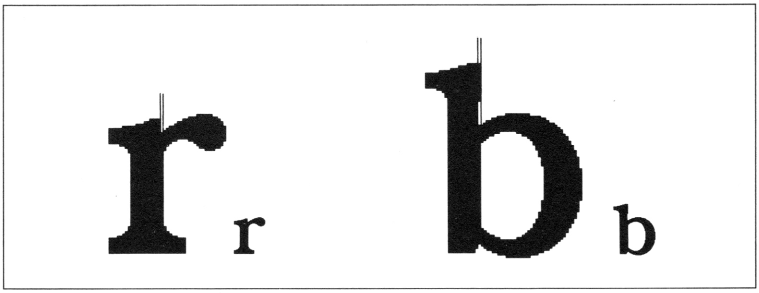


Fig. 116
Deliberate differences are preserved.

7.2.7 Tails

Rastering can sometimes destroy fineness of detail in tails, causing them to disappear completely. When this happens, nothing other than professional manual correction can hope to recreate a fineness of detail to the character.

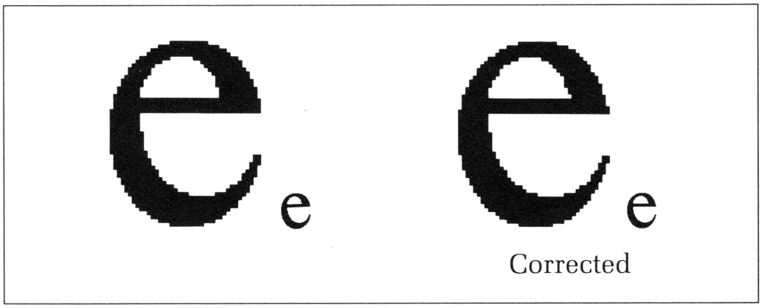


Fig. 117
Small changes sometimes have a great effect.

7.3 SC formats for laser printers

There is no doubt that laser, LED and electroerosion printers will continue to proliferate throughout the graphic arts industry well into the future. They have moved well beyond their original role as «fast printers for computers» in commercial applications. They will certainly increase their use as in-house and proofing printers and are certainly increasing rapidly in connection with PCs as «typewriters» in offices. The key factor in their use has been their considerable price, but as of today, a true 600 × 600 dpi machine handling 11 × 17 inch paper is available for under $4,000. The evolution of various printer engines keeps the quality moving upward and prices steadily moving downward. It is entirely conceivable that by the middle of the 90s decade resolutions greater than 1,000 dpi will be seen in machines selling for under $3,000 as such machines have already been announced.

Most of these printers are classified as «low resolution» devices, writing with 12 lines/mm to 24 lines/mm (300 to 600 dpi). Those developing type for these machines are aware of special consideration required when used at these resolutions (ref. Bigelow).

Poof printing and DTP

Parallel to new typeface development, existing typefaces must be adapted for these printers. Provided these printers are used mainly as proofing devices for high resolution machines, a certain loss of quality is acceptable. There are those publications and applications which have chosen to use lower resolution originals without going to higher resolution, and as laser device resolution increases, the practice is spreading. With toner powder technology a severe limiting factor, it is presently questionable whether this printing method can seriously supplant traditional RC photo paper in high resolution imagesetters, though environmental pressures mount for any workable alternative to the toxic chemistry currently used in the industry. When given a role of proofing device, it is important that reproduction quality be sufficient to permit judging all but the most subtle elements whether a single advertisement or a full newspaper page. Generally, set width, pointsize and weight (boldness) must be essentially the same as the final output. Laser printers, as we have said, are now capable of meeting this objective when 12 lines/mm (300 dpi) or greater is their output resolution.

Laser printers most certainly will continue to play an important role in business offices, being used to create, essentially, a final product of text and typography to be read. To this end, it may be necessary to redesign some typefaces. When this occurs, these points should be taken into account:

The most common text size is 9 pt. Body size is then 39 pixels, a cap height of 26 pixels at 12 lines/mm or 300 dpi. In Chapter 2, overhang of circular shapes was discussed, amounting usually 1.5% of the cap height. When capital letters are 26 pixels in height (39 pixels for the em), round forms are either 26 or 28 pixels in height. This calculates to a total cap and base overhang of either 0% or 3.8%, respectively.

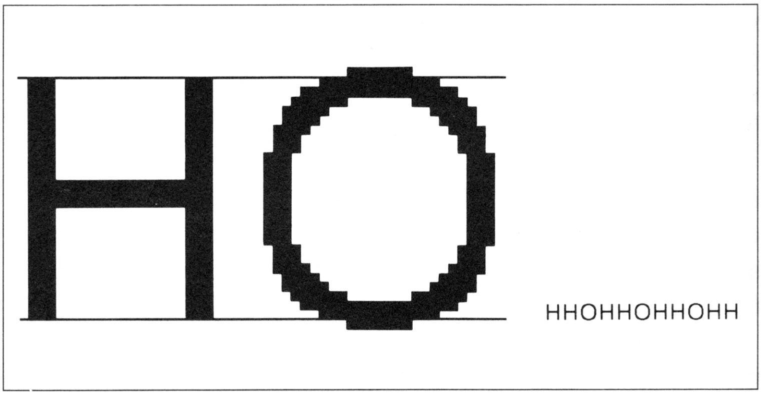


Fig. 118
Raster images
(39 × 39)
with overhang

This large overhang stands out, making the «O» appear larger than an «H», while «O» with no overhang appears too small.

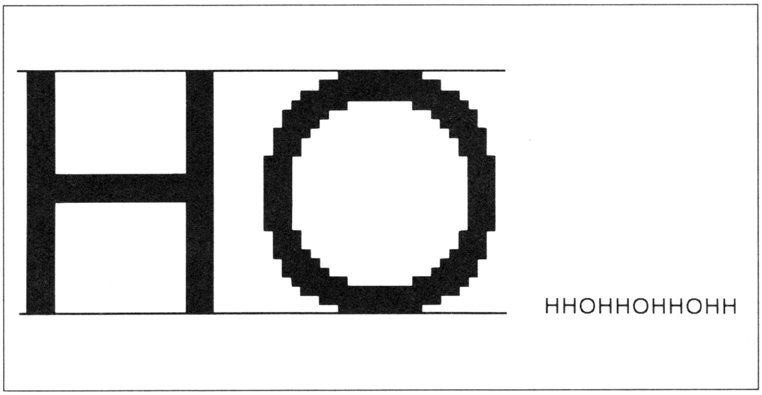


Fig. 119
Raster images
(39 × 39)
without overhang

Half dots, unfortunately, are not available, but half lines can be created with little effort when next to a fully-filled line. A new line consisting of alternating black and white pixels (half-bitting) does the task.

There is no such thing as a half dot.

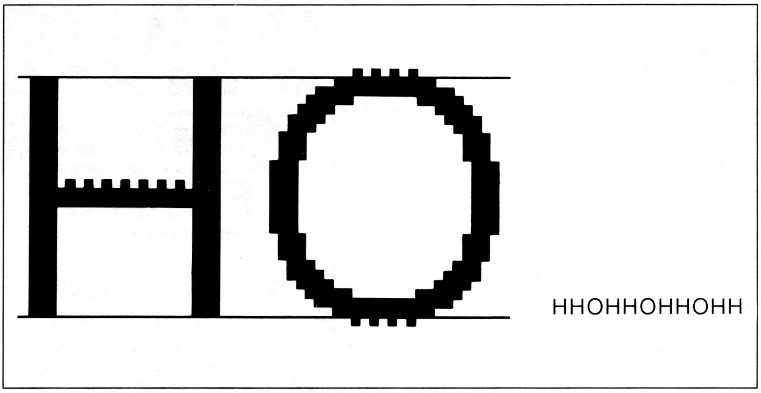


Fig. 120
Raster images
(39 × 39)
with half-bitting

It can be seen on the «H» that its crossbar is 2.5 points in thickness as is that of the «O». The overhang of the «O» now is approximately 1.5%, as it should be. Interesting to note, most laser printers show little zigzagging (or jaggies) when the half-bitting method has been employed to produce stroke thicknesses in half-pixel sizes. A fortunate technological reality for manufacturers of printers is that equalization of charge distribution and distribution of toner often produces an effective blending of optical density consequently lessening the visual distortion which might otherwise be apparent.

The examples shown here show clearly that only machines with writing densities of 24 lines/mm (600 dpi) are consistently capable of producing a correct and aesthetically important overhang. Our recommendation to manufacturers is that only such machines be considered for this purpose.

At least 600 dpi is required for DTP.

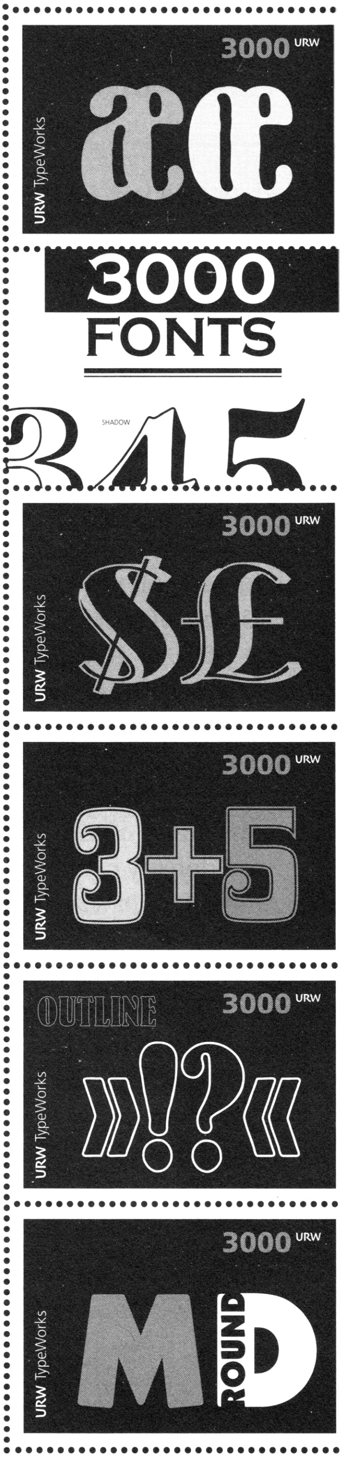
Even when individual dots cannot be reproduced with 24 lines/mm (600 dpi) machines, overall appearance of individual characters is sufficiently fine and pleasing to the eye and type representation is generally of a higher quality.

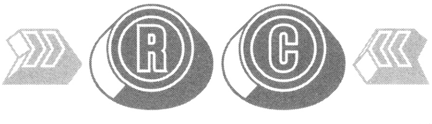
URWTypeWorks ORIGINAL

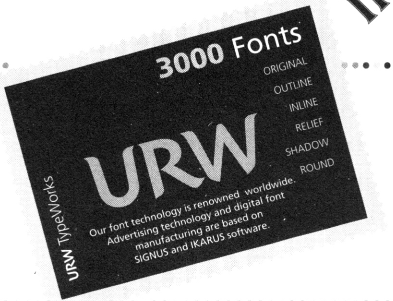
URW TypeWorks assures high-quality setting for graphic design. Each of the 500 basic typefaces also has five variations: outline, inline, relief, shadow and round. That makes 3000 PostScript Type1 or TrueType fonts in proven URW quality **unlocked** on a CD-ROM for the Macintosh or PC. All fonts have unique kerning tables with more than 1000 kerning values. And you will also get **Kernus** on the TypeWorks CD. Kernus is the program for automatic production of pointsize related kerning tables. The typefaces are selected from the international bestseller lists from the last few years. Among them are the newest designs from ITC and internationally known typeface designers. We are offering you URW TypeWorks together with a **multisession CD-ROM drive**.

RELIEF

INLINE


3000 URW
æœ
URW TypeWorks
3000
FONTS
345
SHADOW
3000 URW
$£
URW TypeWorks
3000 URW
3+5
URW TypeWorks
OUTLINE
3000 URW
»!?«
URW TypeWorks
3000 URW
MD
ROUND
URW TypeWorks





3000 Fonts

ORIGINAL
OUTLINE
INLINE
RELIEF
SHADOW
ROUND

URW

Our font technology is renowned worldwide. Advertising technology and digital font manufacturing are based on SIGNUS and IKARUS software.

URW TypeWorks

URW Software & Type GmbH
Harksheider Straße 102
D-22399 Hamburg
Tel. +49 (40) 60 60 52 70/71
Fax +49 (40) 60 60 52 52

Industry Standards

Today's computers store fonts as outlines and equip them with instructions (hints) for intelligent sizing (scaling). This technique employs a single format to generate text on display screens, laser printers, drafting devices and typesetting machines. Because the latter two, typesetting and drafting devices output at high resolution, only outline information determines their final output quality. In contrast, instructions play a decisive role in delivering quality reproduction on display screens and laser printers. Only when coupled with instructions do outline data produce optimal images in bitmap form, irrespective of resolution capabilities of screens and printers.

Hints,
or instructions

Scaling techniques occupy a good number of companies, most significant among them, Adobe with PostScript, Agfa Corp. with Intellifont, Apple Computer and Microsoft with TrueType, SUN/Folio with Fontscaler, and URW with NIMBUS.

List of instructions

Apple Computer coined the term «instructions» while Adobe Systems uses the designation «hints». Other companies simply speak of intelligent fonts or «intelligent outlines». Although «switches» have long been used at URW, we will use the term «instructions» in future discussion. The word's Latin derivation seems to make «instructions» suitable to communicate the reality of a requirement of artifical intelligence in font scaling programs so that raster image processors (RIPs) function optimally. It is correct to consider fonts as being comprised of descriptive elements. Straight vertical stems are doubtlessly the primary elements in Latin alphabets, with crossbars as found on «E» and «H», next in importance.

The key elements that describe a typeface

Arches found in «O», or the curved connections in «h», «m», and «n» are important elements as are the uniquely sculpted «feet» terminating most stems in serif style fonts. The Times Roman letter «I» exhibits a vertical stem with serifs at either end. Another special element is the diagonal stem. Various other determining components describe the remaining letters in a font, so numerous it is helpful to list them and their characteristics. These are the instructions which are used by scaling programs to reproduce fonts optimally, particularly at small sizes (10 pt or smaller) and output at low resolution (100 to 300 dpi). The following, lists the assigned name and corresponding symbol or icon of each instruction followed with brief comments.

Icon	Instruction	Comment
▮	Stem	Keeps stroke widths of stems, defined by two straight, vertical borders, consistent
▬	Bar	Keeps width of bars, defined by straight horizontal borders, consistent
(	Bow	Keeps width of bows, defined by two round, vertical borders, consistent
◠	Arch	Keeps width of arches, defined by two round, horizontal borders, consistent
[	Curve stem	Keeps width of vertical stems, defined by one straight and one curved border, consistent
⊓	Curve bar	Keeps width of horizontal strokes having one straight and one curved border consistent
□	Counter	Keeps width of white spaces (counters) consistent
‖‖	Weight	Maintains canonic consistency of related stroke widths
/	Slant	Keeps width of diagonal strokes consistent (diagonals)
⌖	Extreme	For placement of curve points lying on extremes (deepest, highest, furthest left or right control points)
⊥	Serifs	Controls horizontally drawn serifs and partial serifs
⊣	Bar serif	Controls vertically drawn serifs and partial serifs
⊖	Overhang	Uses baselines to control arch overhangs
⧖	Tension	Straightens shallow curves at small pointsizes (Optima-switch symbol)
◐	Spot	Adjusts stroke width for white or black writing engines
△	Delta	Special instructions in TrueType for maintaining a minimum width for strokes or for inserting single pixels
( )	Dropout	Maintains a minimum width for strokes

Fig. 121
Icons and names for different instructions (hints)

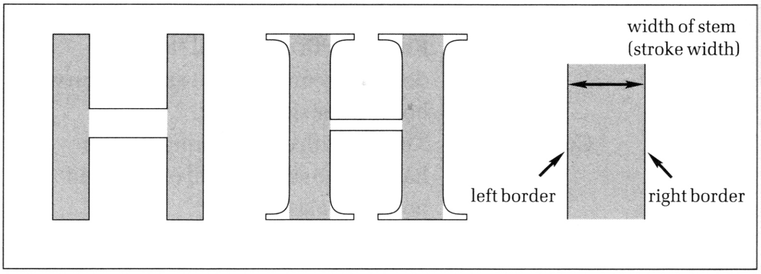
Examples of instructions

The most important instructions, those with horizontal and vertical strokes, are selected as an example for more detailled explanation.

Stems

Stems consist of two straight vertical borders (B, D, E, F, H, I, etc.). «H» has two stems, both of which may be terminated in serif style font with serifs or feet at either end.

Fig. 122
Stems are described by these instructions:
- *stem class (uppercase, lowercase, etc)*
- *x position of a border*
- *x stroke width*



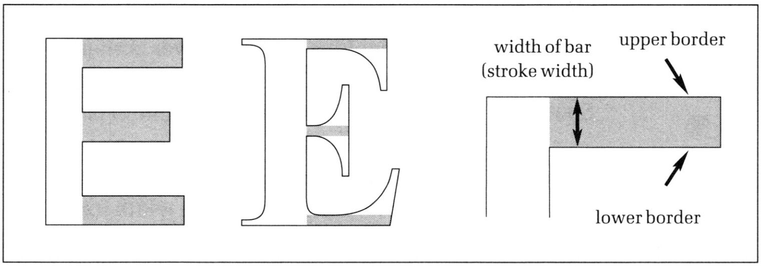
Considered as an element «added on», serifs are treated as though they were merely pasted onto a stem, regardless of their particular size or shape.

Bars

Bars are distinguished by instructions analoguous to those for stems:

A bar is a straight, horizontal stem bordered by two straight horizontal lines (E, F, H, L, etc.). Cap «E» is a classic example, with three bars. Serifs often terminate bars, also, and are called «bar serifs».

Fig. 123
- *class*
- *y position of a border*
- *stroke width*
- *or relative y position of the other border*



8.1 Type1

by Jochen Lau

PostScript language technology with their so-called Type1 format began to make themselves known as early as 1985, largely as the fonts in ROM of the Apple LaserWriter. This was the first successful laser device to receive its page description data in a PostScript format, and with the help of a raster image processor (RIP), calculate the necessary bitmap representation for page output. PostScript is a programming language developed by Adobe Systems for description of graphics. An interpreter within the RIP takes incoming PostScript statements and converts them into output data. In addition to the large number of language elements (math and logical operations) making up all programming languages, PostScript also incorporates graphic operators to draw elements such as lines and curves, or to fill areas with color.

PostScript – a programming language for type and graphics

A key component of virtually every graphic is text or words, whether it be headlines, part of a logo or even continuous text on the page. To handle text, PostScript provides a «font construct», with which outlines of characters in a font are described using straights, Bézier curves and other special operators to maximize efficiency of text output. In the PostScript world, font «programs», rather than font «designs», are spoken of, mainly because American copyright law protects computer programs but not font designs.

Special data types and commands for text

PostScript uses several font coding methods. Type3 format has been in the public domain since inception. Utilizing language elements already published by Adobe Systems, users were able to generate fonts which produced good results on high-resolution output devices. These Type3 fonts were lacking inbuilt «intelligence» to eliminate any undesirable side-effects of rasterizing at low-resolution (see Chapter 7).

Various typeface formats

Type3 without hints

To alleviate such problems, Adobe Systems had developed a solution when it first introduced PostScript language. This format included what they called «hints» to mark specific character elements for special treatment by the rasterizer during bitmap generation. This coded format, known as Type1, was kept a closely guarded secret by Adobe for many years, using it exclusively for their own manufactured fonts. Not until serious challenges from rival developers threatened

Type1 with intelligence

to blow the secret open, were the Type1 specifications made known publicly (Lit. AD 90). This revelation has transformed PostScript and Type1 fonts into the most commonly used of all standards in the world of computer typesetting and publishing.

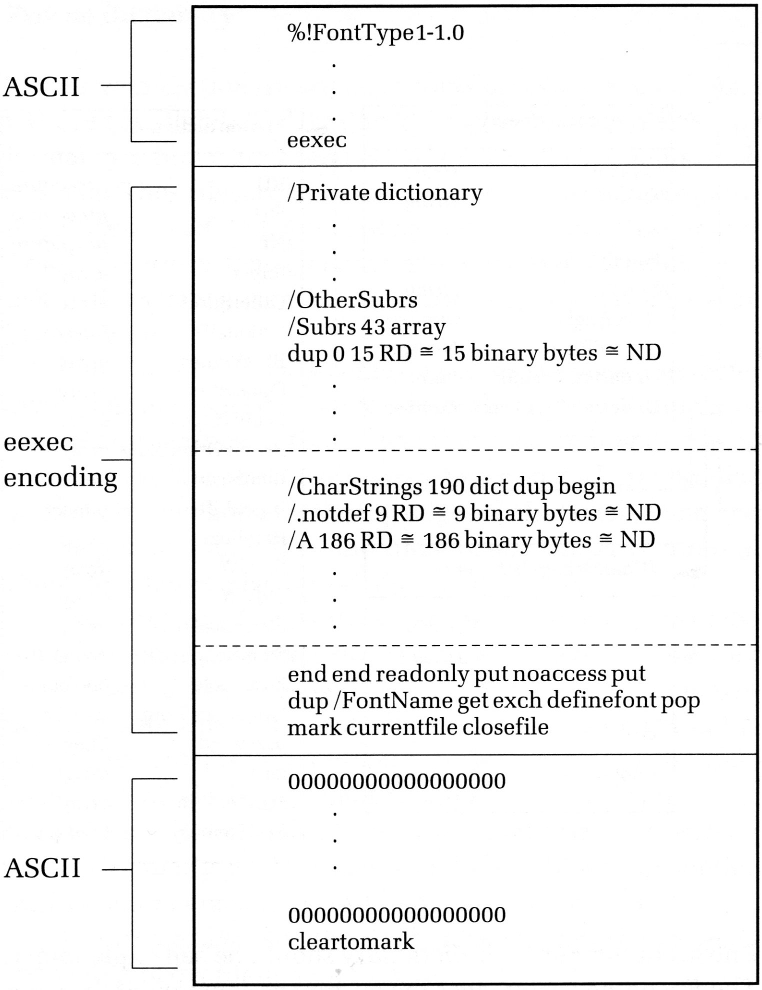
This chapter describes the general structure of PostScript Type1 fonts, detailing the mathematical description of character geometry and technical requirements the outlines must maintain. We also survey data (hints) used in intelligent scaling. If even more detailed information is desired, refer to the description of Type1 format stated earlier or the PostScript Reference Manual (Lit. AD 91).

Structure of a PostScript Type1 font

To describe a PostScript font, a special dictionary, the *Font Dictionary*, must first be generated. This kind of dictionary contains tables used to assign variable values to keyword operators. The bulk of entries pertains to both Type1 and Type3 formats.

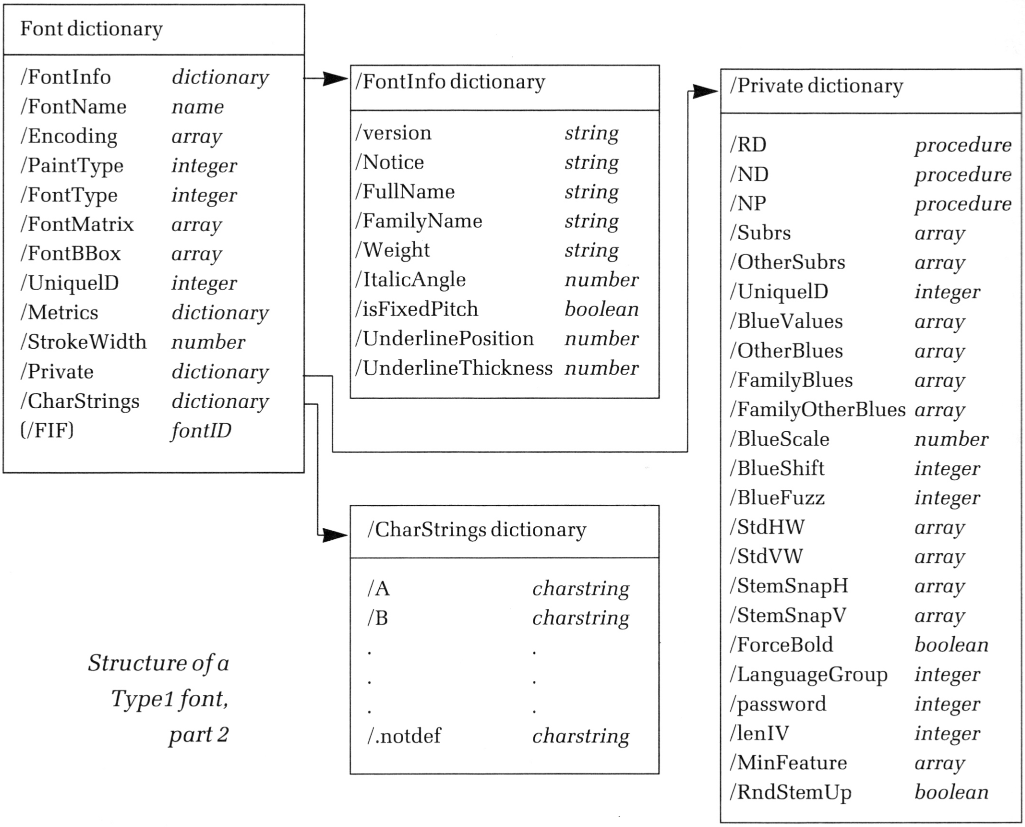
Fonts are coded.

As a rule, PostScript programs are written in a readable form, in text or ASCII form. An exception, however, is Type1 code, where the FontInfo dictionary and a portion of the Font Dictionary are written in ASCII text but are followed by *eexec* operators, denoting starting points for coded *CharStrings* and *Private* dictionaries. The next two figures demonstrate the structure of a font.



```
%!FontType1-1.0
      .
      .
      .
eexec
/Private dictionary
      .
      .
      .
/OtherSubrs
/Subrs 43 array
dup 0 15 RD ≅ 15 binary bytes ≅ ND
      .
      .
      .
/CharStrings 190 dict dup begin
/.notdef 9 RD ≅ 9 binary bytes ≅ ND
/A 186 RD ≅ 186 binary bytes ≅ ND
      .
      .
      .
end end readonly put noaccess put
dup /FontName get exch definefont pop
mark currentfile closefile
0000000000000000
      .
      .
      .
0000000000000000
cleartomark
```

Structure of a Type1 font, part 1



Font dictionary	
/FontInfo	*dictionary*
/FontName	*name*
/Encoding	*array*
/PaintType	*integer*
/FontType	*integer*
/FontMatrix	*array*
/FontBBox	*array*
/UniqueID	*integer*
/Metrics	*dictionary*
/StrokeWidth	*number*
/Private	*dictionary*
/CharStrings	*dictionary*
(/FIF)	*fontID*

/FontInfo dictionary	
/version	*string*
/Notice	*string*
/FullName	*string*
/FamilyName	*string*
/Weight	*string*
/ItalicAngle	*number*
/isFixedPitch	*boolean*
/UnderlinePosition	*number*
/UnderlineThickness	*number*

/CharStrings dictionary	
/A	*charstring*
/B	*charstring*
.	.
.	.
.	.
/.notdef	*charstring*

/Private dictionary	
/RD	*procedure*
/ND	*procedure*
/NP	*procedure*
/Subrs	*array*
/OtherSubrs	*array*
/UniqueID	*integer*
/BlueValues	*array*
/OtherBlues	*array*
/FamilyBlues	*array*
/FamilyOtherBlues	*array*
/BlueScale	*number*
/BlueShift	*integer*
/BlueFuzz	*integer*
/StdHW	*array*
/StdVW	*array*
/StemSnapH	*array*
/StemSnapV	*array*
/ForceBold	*boolean*
/LanguageGroup	*integer*
/password	*integer*
/lenIV	*integer*
/MinFeature	*array*
/RndStemUp	*boolean*

Structure of a Type1 font, part 2

Entries in the *FontInfo dictionary* should be self-explanatory. The *Encoding* vector gives the current assignment of the character code in text to character names used by the PostScript interpreter; enabling characters to be located in the CharStrings dictionary. The *FontType* must, of course, be 1. For Type1 fonts, the *FontMatrix* (a bounding box large enough to enclose each character in the font) is used for calculations and estimations and consequently should be as precise as practicable. The *Unique ID* is the number the interpreter uses for internal font access operations. This number must be unique within this system to avoid serious conflicts. *PaintType*, together with *StrokeWidth*, determine whether the characters will be outline or filled with color. Private and CharStrings dictionaries are explained next.

Private Dictionary

Global units

The Private Dictionary contains hints applying to all characters in a font. Similar to URW's NIMBUS technology, the Adobe format uses hints describing specific character elements, such as stems. These descriptions are evaluated during the rasterizing process itself. Other techniques use a different approach: TrueType, for instance, contains extremely detailed element instructions telling the rasterizer how to place every single dot in an outline.

Overhang suppression with BlueValues

BlueValues and OtherBlues, used to suppress character overhangs in small-sized fonts, are made of pairs of numbers, each pair describing a font line and corresponding overhang zone. *BlueValues* contain overhang zones on the baseline, plus up to six further zones at the top edges of characters (x-height, cap height, ascender line and numerals). Overhang zones at character's lower edges are given in *OtherBlues*.

FamilyBlues ensures uniform appearance of a font family

Text copy nearly always uses more than one weight of the same font family. Because this is true, it is important that font lines of each different weight be assigned the same values, particularly with small size fonts. If, for instance, x-height in the normal weight font is computed to be 5.4 pixels, the bold weight 5.6 pixels, both should be adjusted to the same value, say 5 pixels. The two fields *FamilyBlues* and *FamilyOtherBlues* operate similarly to BlueValues and OtherBlues and define overhang zones occuring most frequently in a font family.

The *BlueScale* and *BlueShift* values permit size-dependent control of overhang suppression.

Standard stem widths for the various character classes

To retain a balanced and even appearance at small sizes, vertical and horizontal stems must be kept to a uniform width. Upper and lowercase characters probably have different widths for vertical stems. It is possible for stem widths to deviate from a mean value within a class of characters, having been caused by fault in design or possibly statistical errors arising when data was generated.

Were each stem rasterized in isolation, its edges would remain aligned to pixel borders, a 1.4 pixel stem would be rounded down to 1 pixel, whereas a 1.6 pixel stem would round up to 2. To overcome this situation, it is possible to define as many as 12 preferred stem widths in the fields *StemSnapH* and *StemSnapV*. These values denote the average stem widths in each of the character classes. In smaller fonts, *StdHW* and *StdVW* become operative and all stems are adjusted to the correct values.

Again, in small pointsizes, particularly for screen display, medium and bold weight typefaces might both use 1-pixel wide strokes, making them indistinguishable in a bitmap format. Yet another form of hint, *ForceBold*, guarantees that two weights can be distinguished, even at small sizes.

Adobe uses the *Language Group* parameter to divide fonts into two categories: group 0 (zero) for Latin, Greek and Cyrillic, and group 1 for Chinese, Japanese, or other non-Latin type. Though it could be useful for the rasterizer to know which of the two groups a font belongs to, in actuality, no specific effect is noticeable using the description.

Two other important entries in the Private Dictionary are *Subrs* and *OtherSubrs*, which will be examined in CharStrings dictionary section.

Other Private Dictionary operators shown in the table on page 146 have no effect. Some must be specified for compatibility matters, while others need no longer be used.

CharStrings Dictionary

Paths create outlines.

This is where character outlines are described. The same as all other graphic objects, they are represented by paths consisting of straights and Bézier curves and can include numerous closed contours. Outlines are described with a subset of PostScript graphic commands, able to call Private Dictionary subprograms for descriptions when character elements occur more than once. The CharStrings Dictionary also stores character-specific hints used to optimize rasterized characters.

Characters are stored in an em square of 1000 × 1000 points.

Characters in a PostScript typeface use their own unique coordinate system, wherein paths are described. As a rule, outline coordinates use a 1,000 × 1,000 unit per em square, with integers only. As some characters, such as mathematical symbols, for example, protrude beyond the edges of the body, Type1 fonts permit coordinates from – 2,000 to + 2,000 units, a resolution Adobe has assumed to be adequate for typeface reproduction. If a typeface producer wishes to surpass this value, the *div* operator can be used to generate fractional values.

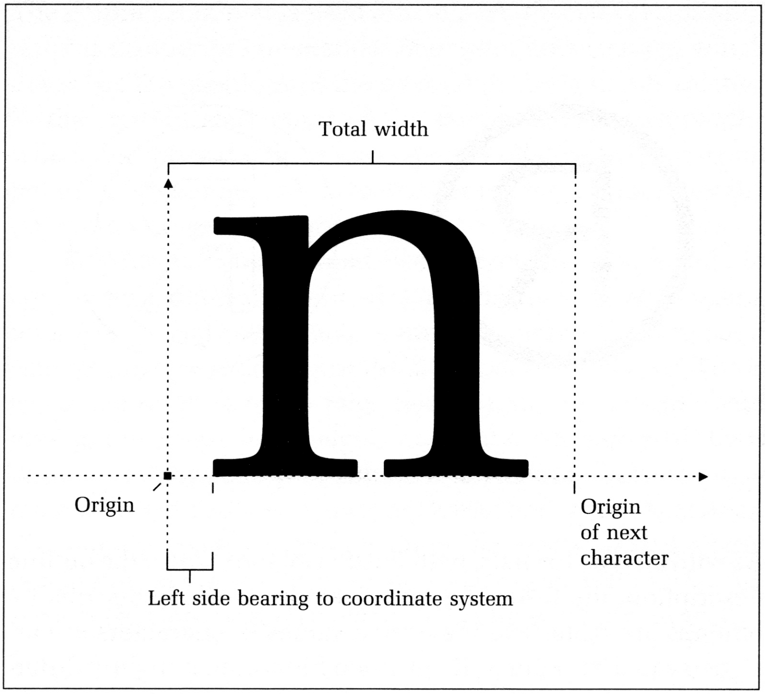


Fig. 124
Coordinate system for PostScript fonts

Conversion of em square to a user system unit

The *FontMatrix* is used to convert a character system into a user system. The matrix generally contains values [0.001 0 0 0.001 0 0] reducing character size by a factor of 1,000 (1,000 character units become 1 user unit or approximately one pica point). The *Scalefont* operator, which computes the correct font size, will thereby have a standardized effect. It is possible to deviate from the Standard-Font Matrix to generate italic, condensed or expanded versions of fonts which originally were within the standard matrix.

When the PostScript interpreter is rasterizing, it assumes the filled size of characters is always to the left of the outline (seen from the direction of movement). This necessitates digitization of the outer contour in an anti-clockwise direction with inner contours reversing direction at each succeeding level inward.

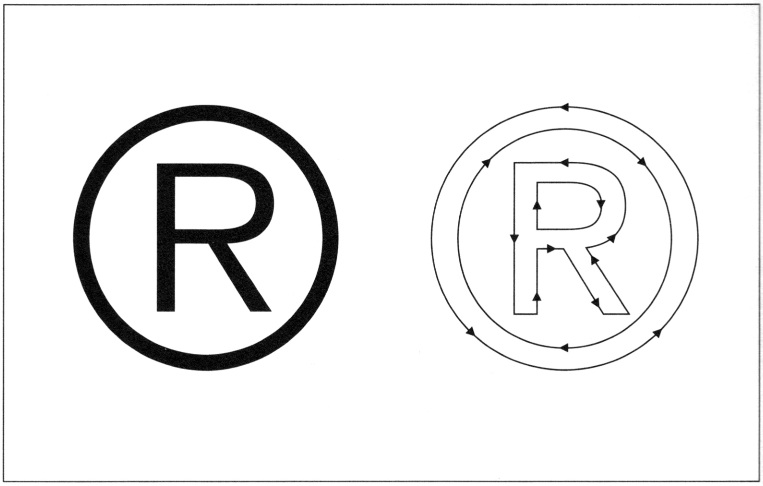


Fig. 125
Outline direction

Technical optimization of font data gives better results.

As with all type formats, with PostScript the shorter the outline description, the faster the rasterizer can work. Better quality bitmaps are rasterized if extreme values of characters are assigned «anchor points». Even when hints adjust slightly different stem widths to the same number of pixels below specific pointsizes, Adobe, and others, recommend making certain of charactering elements, identical in terms of design likewise identical numerically, preventing problems with rastered results.

Characters in a PostScript typeface are rasterized by the *BuildChar* routine. Type3 fonts contain this routine as part of the font. Depending upon requirements of the font, a typeface producer can code such a routine as a PostScript program, enabling all PostScript language elements to be used in character descriptions. A Type1 font, on the other hand, implicitly calls a BuildChar routine built into the interpreter. While this routine only interprets a subset of PostScript commands, it compensates by understanding some commands not actually part of the standard language, e.g., hints.

To save storage space, data, numeric values and commands are stored as integers in binary form and organized in bytes. Certain values carry a defined meaning allowing the interpreter to distinguish between numbers and commands.

Special commands in Type1 format

The graphic commands used are, in part, special variations of normal PostScript commands. *Rcurveto*, for instance, which describes the positions of the next three curve points relative to the current start point of the curve, becomes *rrcurveto*, which describes the three curve points relative to the current point. This technique leads to smaller numeric values thereby reducing storage requirements.

Hint commands for stem control

PostScript Type1 format uses direct hint commands to control horizontal and vertical stems (*hstem* and *vstem* operators). For special cases, such as «m», the operators hstem3 and vstem3 can be used to adjust the stem and white spaces. These stems are defined by an edge position and their width, and may not overlap. Serif heights are also controlled with their stems. Another stem hint function is ensuring correct alignment of upper and lower parts of characters to their font lines. The same approach is taken to align characters lacking a horizontal stem at the required position (Helvetica «**I**» for instance). A so-called Ghost stem is used to aid in aligning the character.

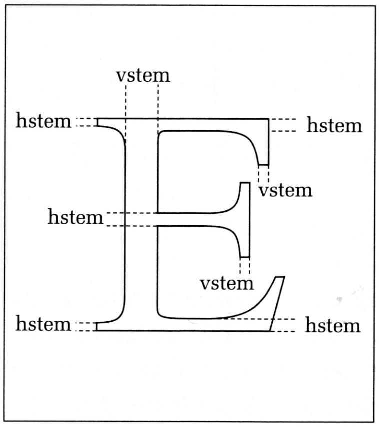


Fig. 126 (far left) Horizontal and vertical stems

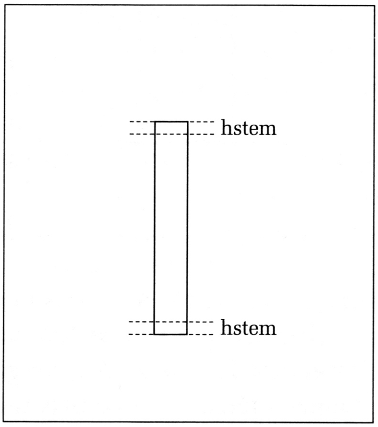


Fig. 127 (left) Ghost stems

Sub-routines to implement further hints

At the opening of this section sub-routines were mentioned which, though stored in the Private Dictionary, are called and executed during character processing. There are two fields of sub-routines, the *Subrs* and the *OtherSubrs*. The first four routines in each of these two groups have a standardized function for all Type1 fonts, indicating that these four routines are no longer evaluated by more recent rasterizers from Adobe Systems. Instead, the call is handled by a fixed section of program code within the Adobe Type Manager or ATM.

These four defined sub-routines implement the Hint Replacement and Flex functions. Other entries in Subrs are available for use by type producers to store serif forms occurring repeatedly in the typeface. OtherSubrs are reserved for extensions.

Hint replacement for overlapping stems

What exactly is hint replacement? As stated earlier, stems which are defined by vstem or hstem may not overlap. The «E» below has three horizontal stems controlled by hstem. The top and bottom serifs are also controlled by hstem but have a different height. The two area must therefore be handled separately. The hint replacement feature instructs the rasterizer to «forget» the first set of hints once it has processed the three stems. The OtherSubrs routine is then called, and switches to new values.

Fig. 128 (right) Hints for first part

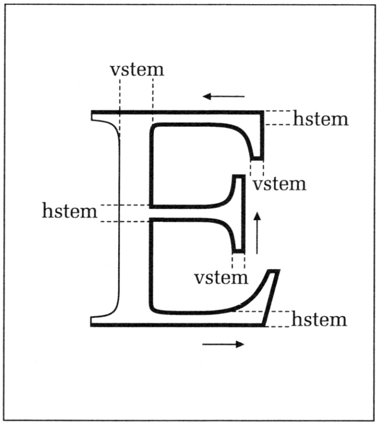
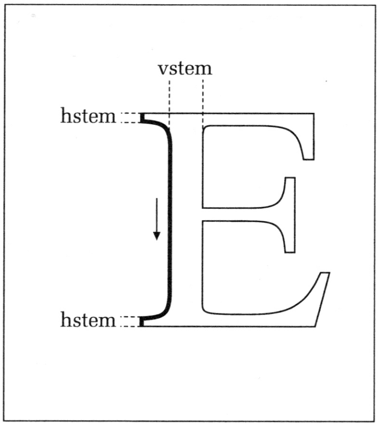
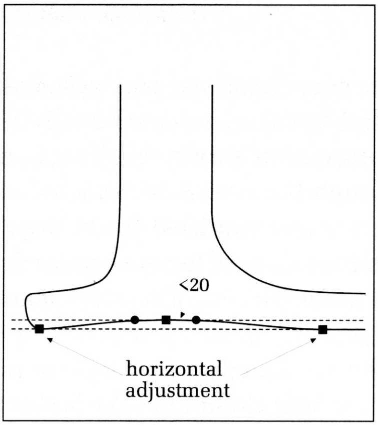


Fig. 129 (far right), Hints for the second part of the character



In small-size fonts, shallow curves running almost horizontally or vertically can appear too large compared to the remaining characters. It may be advantageous in such small-size fonts, to replace such curves with straights, a task performed by the Flex mechanism with the aid of OtherSubrs and Subrs 0 to 2. For the function to work properly, section of the outline to be straightened must observe strict requirements and restrictions in terms of its description with Bézier curves.



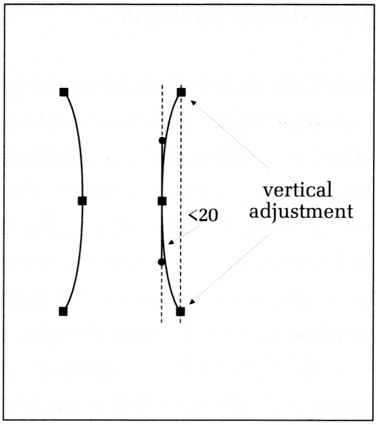


Fig. 130 (far left)
Horizontal adjustment

Fig. 131 (left)
Vertical adjustment

8.2 Intellifont

by Jochen Lau

Another of the four standard «intelligent font scaling» methods described in this book is Agfa Compugraphic's Intellifont System. As the native format of the Hewlett Packard Laserjet III printer series, the Intellifont System is possibly the most widely used in terms of actual installations.

Strictly speaking, «Intellifont» is in reality the rasterizer software generating character bitmaps or outlines in various resolutions and sizes using FAIS format fonts. FAIS (more precisely, Intellifont FAIS) is an abbreviation for Font Access and Interchange Standard, the format in which fonts are supplied for use under PCL5, the Laserjet III printer page description language and other PC-based applications. The term «Intellifont» is, however, commonly used to denote both rasterizer and fonts, no distinction made between the computer program and fonts in this format.

Intellifont = rasterizer + font format

This section introduces the mathematics of character geometry, plus information on outlines and metrics. Instructions used in intelligent scaling are also detailed along with an overview of files supplied on font disks. Readers should consult relevant Agfa Compugraphic brochure CG91 for greater detailed information on the format's «bits» and «bytes».

Coordinate Systems

Kerning in relative units

Intellifont FAIS fonts use two coordinate or unit systems. Kerning values are given in «relative units», obtained by dividing the em square into 54 sections on the X axis, resulting in a 54 unit per em character. Although the actual kerning value remains constant while characters are modified to fit target grids and resolutions, absolute dimensions of the em square do change, so that absolute correction values for character spacing are adjusted accordingly.

«Artwork» = oulines of characters

The size of a character outline – the artwork – is given in so-called «Design Window Units». Agfa Compugraphic's standard character artwork size is 250 points. In this case, one DWU equals one hundredth of a millimeter, the standard of the IKARUS system, used also by Agfa Compugraphic.

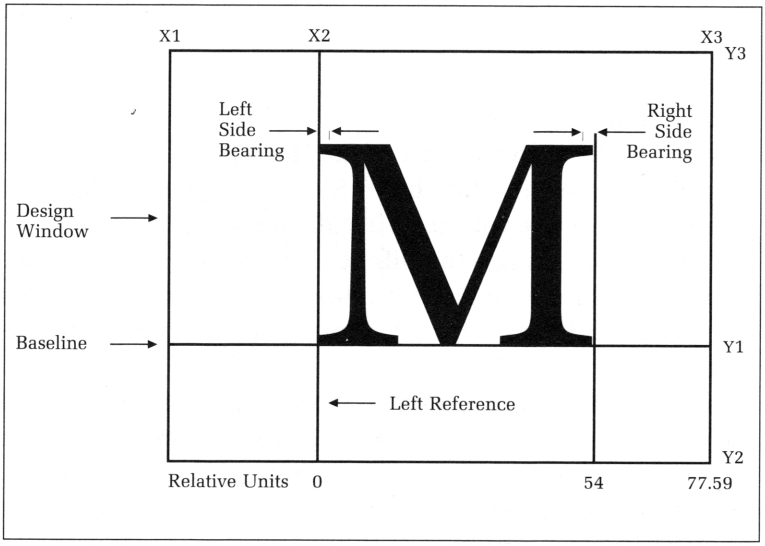
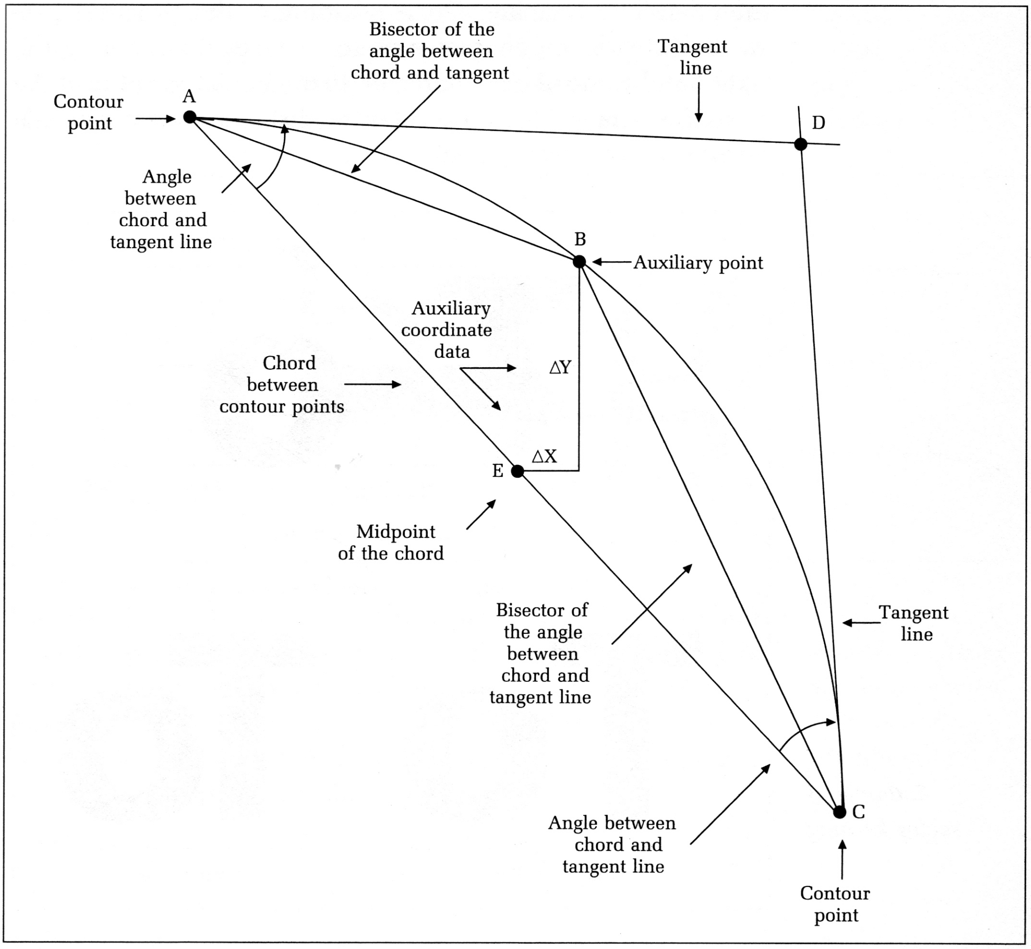

X1
X2
X3
Y3
Left Side Bearing
Right Side Bearing
Design Window
Baseline
Y1
Left Reference
Y2
Relative Units
0
54
77.59

Fig. 132 Design window coordinates and relative

Outline Geometry

Agfa Compugraphic's font storage method is based on the IKARUS format, i.e. font data begins as a series of corner, curve and tangent points. Curves must be transformed into circles to be used in FAIS format (see section 6.3). A control point computed using the Saur methods is stored in with IK points, its position is given relative to the center points on link lines between IK points. Values are restricted to one byte (+128). The figure below illustrates this method, showing also what data is stored in the FAIS format. A short mathematical calculation (not included) demonstrates that this storage technique is equivalent to the DI or VC format.

Fig. 133
Determing control points


Bisector of the angle between chord and tangent
Tangent line
Contour point
A
D
Angle between chord and tangent line
B
Auxiliary point
Auxiliary coordinate data
ΔY
Chord between contour points
ΔX
E
Midpoint of the chord
Bisector of the angle between chord and tangent line
Tangent line
Angle between chord and tangent line
C
Contour point

Kerning

An Intellifont FAIS format font may contain as many as two kerning tables, one for kerning of text and the other for display. Although most fonts normally have a single table (conforming to their design), some are equally well used as text and display. Both display and text tables are structured identically.

Intellifont uses sector kerning

Kerning FAIS fonts is accomplished with what is called «sector kerning». Unlike other methods, such as in PostScript, sector kerning places no upper limit on the number of correction values which may be applied.

Each Intellifont character is divided into horizontal zones or sectors. Within each zone, left and right sides of characters are assigned values constituting potential correction values of intercharacter spacing. Once positioned, the program proceeds through the character sectors one-by-one, adding the right-hand value of one character to the left-hand value of the next. The smallest total indicates the distances characters can be moved to correct spacing.

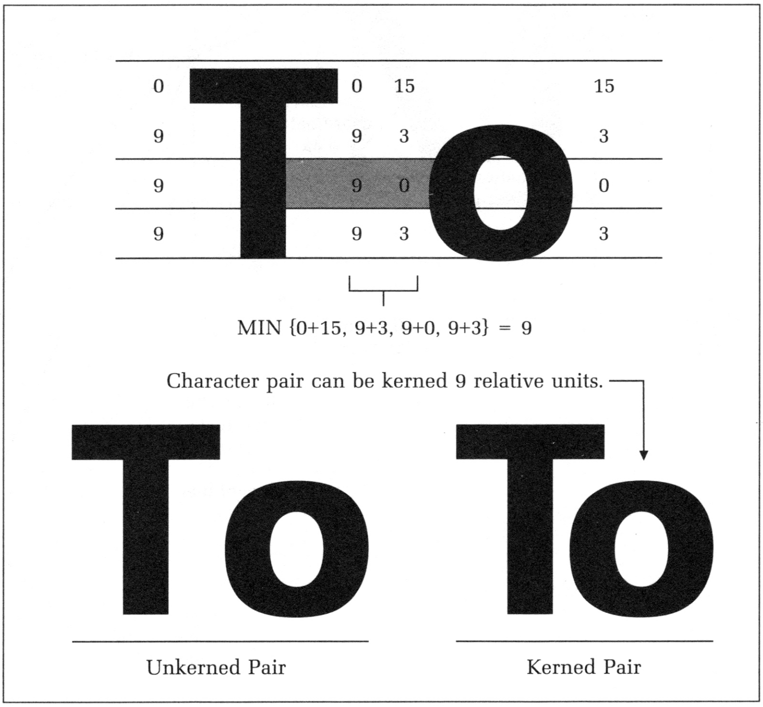

0 0 15 15
9 9 3 3
9 9 0 0
9 9 3 3
MIN {0+15, 9+3, 9+0, 9+3} = 9
Character pair can be kerned 9 relative units.
To
To
Unkerned Pair
Kerned Pair

Fig. 134 Example of sector kerning

Intellifont Instructions

Intellifont makes a distinction between global information (that applying to the entire font) and character-specific information, a feature serving to retain design integrity in instances where an em square spans but a few pixels. As described in section 6.1, the appearance of rasterized characters is enhanced by using additional character information. Agfa Compugraphic refers to this information or data as «instructions».

Font-specific Y classes

We look first at global instructions. A series of horizontal divisions, so-called Y classes, run through all the characters in a font. These classes have «delimiter» lines at top and bottom. The various character elements are assigned to the appropriate Y classes (typically x-height, cap height, ascender height, etc.). Coordinates of the elements need not match precisely the class delimiter lines; allowing for greater flexibility in rasterizing, allowing, for instance, circular overhangs on rounded characters to be suppressed when output at low resolution.

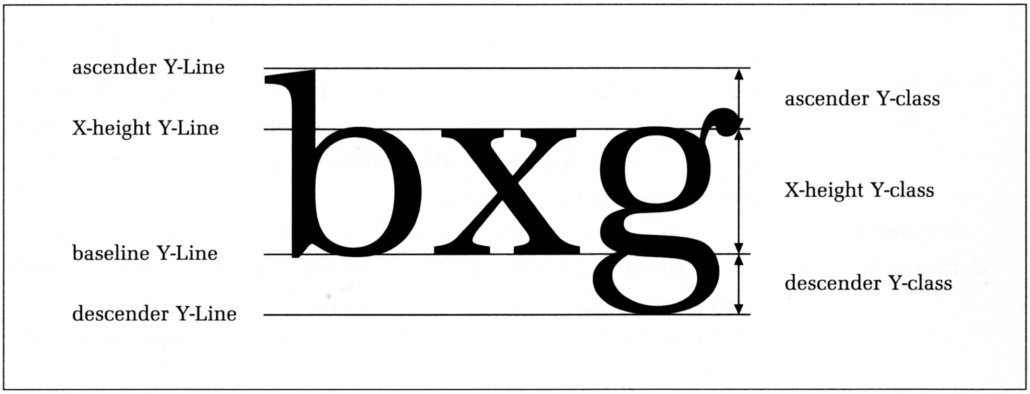


Fig. 135 Y lines and Y classes

Standard unitizing dimension applied with small point sizes.

Standard font dimensions – using separate definitions for both x and y axes – are specified for the font globally. It may be necessary, for example, to thicken curve stems of lower case characters to match the optical width of straight stems. On the downside, the same thickened curve stems might appear inordinately broad at lower resolutions.

Assigning all curved stems an identical standard dimension, we are assured that curved and straight stems are rasterized with the same number of pixels at low resolution.

Typical y axis standard dimensions are assigned as width of horizontal curve stems in:

- lowercase characters (o),
- cap characters (O),
- digits and punctuation (0),
- in superscript and subscript characters and fractals (%).

The following are defined for the x axis as width of vertical straight stems in:

- lowercase characters (n),
- cap characters (H).
- digits and punctuation marks (1).
- superscript and subscript characters and fractals (1/2).

Every character in a font falls into one of these x and y classes. Standard values may be applied when necessary to «level off» slight deviations resulting within each class following rasterizing. A special instruction can also be added to each class, indicating whether its value may be reduced to zero, if necessary, or whether at least one pixel must be retained. Another value, the threshold, indicates the minimum pointsize below which standard dimensions are applied.

Dropout control during rasterizing

Like other rasterizers, Intellifont also provides means to fill in pixel gaps (dropout) as a result of rasterizing extremely thin lines. This feature in Intellifont is called «missing pixel», and can be switched off and on as required using a specific threshold value.

Skeletal points denote «key» point

Turning to Intellifont's character specific instructions, we find that «key» points in characters – their extrema and stem edges, for example – are all marked to ensure optimal character representation. Termed «skeletal points» in Intellifont's format, these points are usually fitted, or rounded, to the character grid during rasterizing.

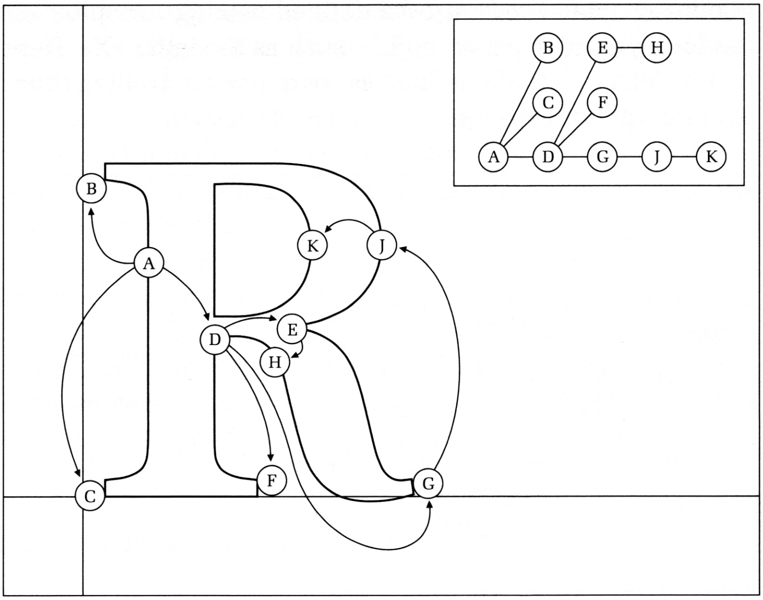


Fig. 136
Skeletal points and associations (x axis)

Particular character features (character width or stem width, for instance) can be defined by specifying associations between skeletal points. These associations are stored in a «tree» structure and processed by the rasterizer in the order they appear in the tree. A suitable choice of associations, coupled with appropriate hierarchical sequencing, allows «preferences» to be taken into account. For example, a preference would cause uniform white space within a character to take precedence over the character's width. Attributes can be integrated instructing the rasterizer to use an association only above a specific pointsize. This feature applied can eliminate ugly stair-stepping or jaggies in shallow curves, such as on serifs in the font Garamond demibold.

«Associations» define how skeletal points are referenced.

«colinear points» to optimize intersections

«Colinear points» may also be defined making for optimized rendering of intersection points, such as the letter «X». Here, two skeletal points are defined as «parents», controlling fitting of critical points to the grid.

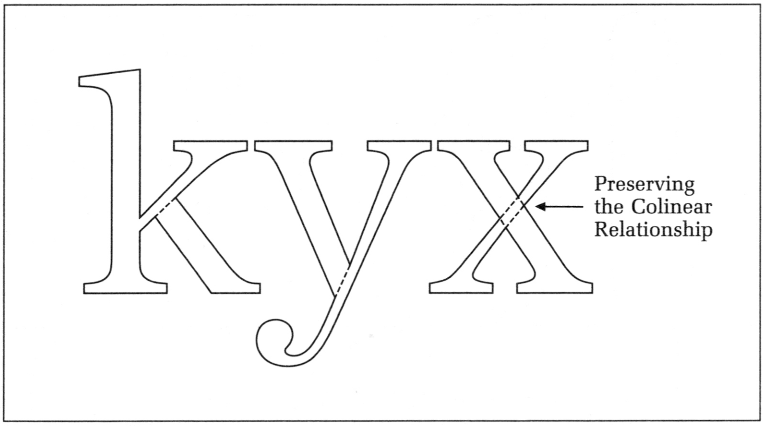


Fig. 137 critical intersection prints

Intellifont FAIS Files

The following descriptions and information files are included on the data medium (computer disk) with the fonts:

- a directory
- a font index
- a font attribute file for each font
- a font display file for each font

The directory contains ASCII-format data with general information concerning data media (size and creation date), name of each font and whether it is designed primarily for text or display applications, and finally, which (if any) kerning table is supplied.

The font index file contains a data set for each font attribute and display file contained on the disk. It holds the font number, file name and character layout, all of which information can be accessed and processed by font downloading programs.

Apart from licensing information, each font-specific, attribute file contains metrics data (maximum ascender and descender length, cap height, x-height and italic angle reference), the exact designated font weight, character numbering information, possibly also application-specific coding data (fontalias), and character spacing values for normal and close setting.

The font display file contains a description of the character artwork and their integrated «intelligence». A distinction is made here between global instructions and those applying only to specific characters.

Attribute and display files are similarly structured to those in other vendor's formats (TrueType). The contents of files is divided into sections, each of which is identified by keyname. At the file header (or start of a section) there are listed key names and section addresses in that file, speeding access to information within the file.

8.3 TrueType Format

by Hartmut Schwarz

TrueType fonts contain detailed instructions for fitting the contours to the output resolution.

TrueType format stores its character images by defining the on-and off-curve points which describe character contours. A relatively straightforward rasterizing program (the scan converter) converts fonts into bitmap format in any pointsize, customized to the resolution of the output device, whether screen, printer, phototypesetter, etc. (Lit. AP 91).

In order to maintain output quality, even at low resolutions, shapes of character contours are adjusted or fitted to the output grid before being rasterized. Information fitting the character to the grid is actually part of the font itself.

Essentially, TrueType format provides a set of extensive and flexible instructions as a framework for the grid fitting process, while leaving sufficient opportunities for producers and developers to implement their own intelligent font scaling schemes. In addition to character data, the TrueType format contains keyboard layout information, spacing, naming, font statistics and other typographic data. In fact, the file structure is flexible enough to accept additional font data at a later time.

8.3.1 How the TrueType File is Structured

The table directory at the start of the TrueType file contains references to the data blocks in the file, the so-called TrueType tables.

Font data in a TrueType file is stored in a series of separate data blocks. The blocks, or tables, are identified by a special four-letter mnemonic code. Character raster image information, for example, is stored in a table called *Glyph Data Table*, designated «glyf».

All access to these data blocks is controlled by a table directory which contains names of all tables in the font, listed under their identifiers. Also included is block length and distance in bytes between file start and the start of the block.

Currently, around twenty table types have been defined, some of them mandatory for the font, other optional. Additional tables may be added without affecting accessibility of existing tables.

8.3.2 TrueType Tables

head, hhea

The *font header table* with the identifier «head», and the horizontal header table with the identifier «hhea», both contain information pertaining to the entire font, such as body size, bounding box, line spacing and administration data (creation date and version number). All data found in the horizontal header table pertains to horizontal layouts (type running horizontally). A *vertical header table*, likewise can be used for fonts set vertically.

name

Carrying the identifier «name», the *naming table* contains the various names required to identify the font (family name, weight designation, complete font name).

glyf

Mentioned earlier, the character data itself is located in the glyph data table with identifier «glyf». For each character in the font exists a separate data block, consisting of outline data, grid fitting and positioning instructions, and, in the case of composite characters (characters comprised of parts of other characters), transformation data. Every character has its specific outline data or contains reference information concerning the other characters from which it is composed (using line transformation such as rotation or sizing). Accented characters are examples of *composite glyphs*. To reduce overall data size, composites are created by simply adding appropriate accent marks to existing basic characters.

The glyph data table contains the characters outline information plus instructions on how the contours are to be fitted to the output raster.

loca

Character block length is variable. The position and length of each block are stored in yet another table, the so-called *index to location table*, with the identifier «loca». This table comprises nothing more than a list of start positions of each character block in a font, arranged in order of their appearance in the glyph data table (the start position of the nth character is located in the nth element of the index to location). Character block length is determined by calculating the difference between a character's start position and the start position of the next character.

cmap

The glyph data and index to location tables are nothing more than a collection of different characters. To enable correct character access for text output, TrueType contains a *character to glyph index mapping table*, identifier «cmap». This table serves as a link between character numbers (ASCII numbers) and the character index showing the position of characters in the index to location table.

hmtx, kern

Character spacing data – left side bearing plus total width of each character – is stored in still another table, *horizontal metrics table* with identifier «hmtx». TrueType fonts may also hold kerning data, stored in a table identified as «kern». A range of formats – allowing greater or lesser tightening – is available for kerning tables.

cvt

Control values, applying to the font as a whole, can be stored in the *control value table*, with identifier «cvt». Values stored here can be accessed during the grid fitting process to ensure uniformity of lengths and size of stems in a font (stem control).

fpgm

The *font program table*, identifier «fpgm», contains instructions executed as a font is first loaded. Generally, these instructions are default settings remaining constant for all pointsizes of the face.

prep

The *control value program table*, with identifier «prep», contains instructions which also are executed as the font is loaded and again when pointsize is altered, and ensure that control values are modified to match current font size.

maxp

Further tables store cross-font data and administration data for character access.

The *max profile table* has the identifier «maxp», defining the maximum quantity of points, contours, instructions, etc. contained in a single font character. This data is essential during font output ensuring error-free interpretation of character data.

post, OS/2

Further tables are required for fonts to be used in specific system environments. When used with a PostScript output device, all necessary specific information is held in a table with identifier «post». Similarly, fonts to be used under MS Windows or the OS/2 operating system use a table accessed with the identifier «OS/2».

8.3.3 Outline Representation

The geometrical path of an outline is represented by a series of on-curve and off-curve (control) points. Characters are digitized from the left, the center of the character located to the right of the start point (see Figure 1).

Curves are represented by a series of parabola sections, defined by on-curve and off-curve control points.

Two consecutive on-curve points are linked by a straight. Curves are represented as a series of parabola sections. Control points (off-curve points) between every two on-curve points, define direction and angle of tangents between on-curve points, determining routing of parabola sections. Tangents are always aligned with the on-curve point and the next off-curve (control) point.

Longer curves may also be coded using an uninterrupted series of off-curve (control) points. When this is the case, the points midway between each pair of off-curve points become actual on-curve points which will be joined up by the parabola sections. In other words, it is at these midway points that the curve touches the line formed by the two off-curve points. This mathematical curve function is given the name quadratic B splines.

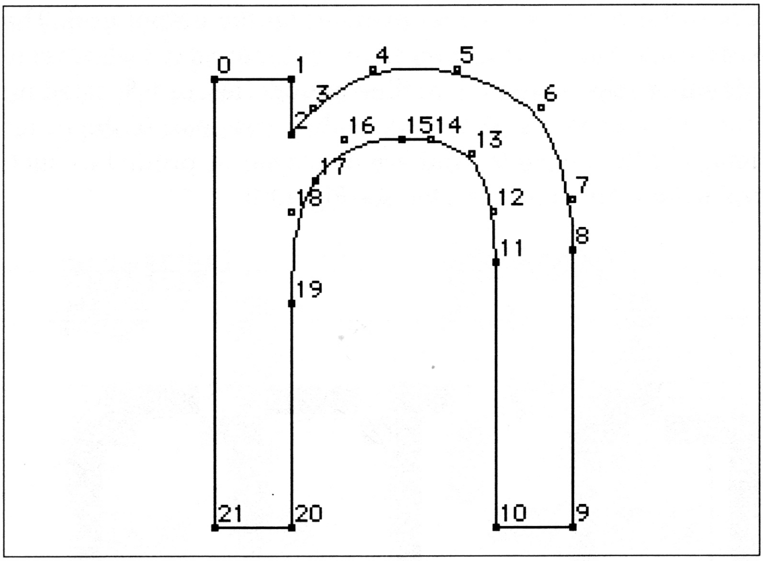


Fig. 138 Outline represented by on-curve and off-curve points.

8.3.4 Scan Converter

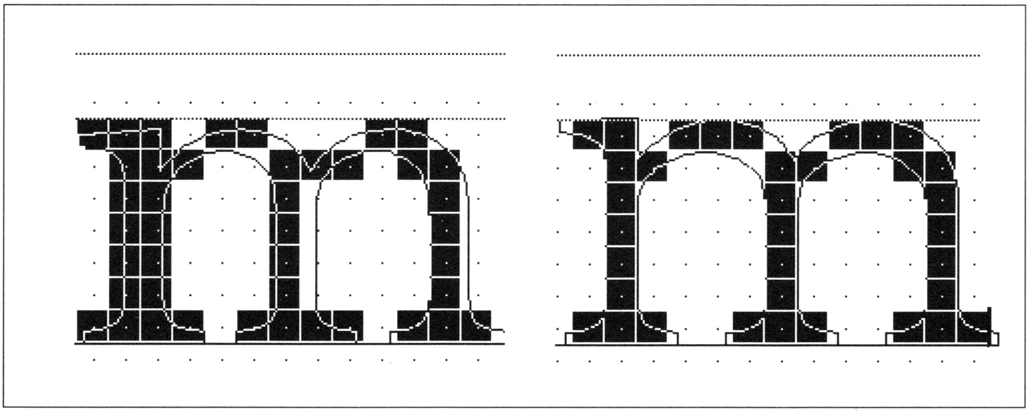
The TrueType rasterizing program mentioned in Section 1, is generally installed in the output device (in ROM) or forms a part of the driver software. At first glance, therefore, it would seem to have little to do with the font format itself.

Contour fitting instructions can only be used properly if the rasterizing program is precisely defined and responds in exactly the same way on every output device.

Any discussion of the TrueType format, however, must describe features and actions of the scan converter. Intelligence of TrueType fonts derives chiefly from the grid fitting programmed by the manufacturer to match the font to a specific output grid. This can only be correctly done when the rasterizing process conducted by the scan converter has been defined as precisely as possible. Font producers can only use instructions properly when assured that the rasterizing program will respond identically on every output device. The manufacturers of these devices must, therefore, adhere to a very strict set of requirements, and may not, for instance, attempt character quality improvement by modifying the scan converter program in any manner. Though such modifications could be advantageous used with certain grid fitting methods, problems will inevitably arise when fonts with a completely different fitting technique – say, one strictly adhering to the rules – are output on the same device.

In general, characters are rasterized according to this rule: Character outlines are first overlaid on the output grid. The scan converter then turns on all pixels (image dots) whose center either falls within the outline boundaries, or is located on one of the contours. These pixels thus form part of the raster image of the character and are displayed or printed as such upon output of that character (see Fig. 139).

Fig. 139
Rasterizing the character «m» with and without grid fitting



Using this simple method, some dropout (gaps) may occur in the raster image, even when a grid fit has been performed correctly, especially true when small pointsize characters, perhaps containing some transformation (rotation, skewing) are output on very low resolution devices. To counter this tendency, the scan converter can be switched to an alternate mode, the scan control mode, via instruction within the font. There are two variations of the mode, allowing extra pixels to be turned on in the raster image (see Figs. 140 and 141).

When not possible with grid fitting, dropouts can be prevented by means of a special scan control mode.

The first variation operates as follows: When a line linking two pixel centers is intersected by two opposite contours, and neither of the pixels has been turned on by the simple method being applied, one of the two is turned on. When pixels are stacked vertically, the lower one is turned on. If positioned side-by-side, the left pixel is turned on.

The second variation applies the same rule, but with this constraint: A pixel is only switched on when opposite contour lines also intersect the three pixel link lines forming a square with the link line under consideration.

Unlike the first mentioned variation, this one closes only genuine gaps within an enclosed area, not turning on any additional pixels at the ends of character sections.

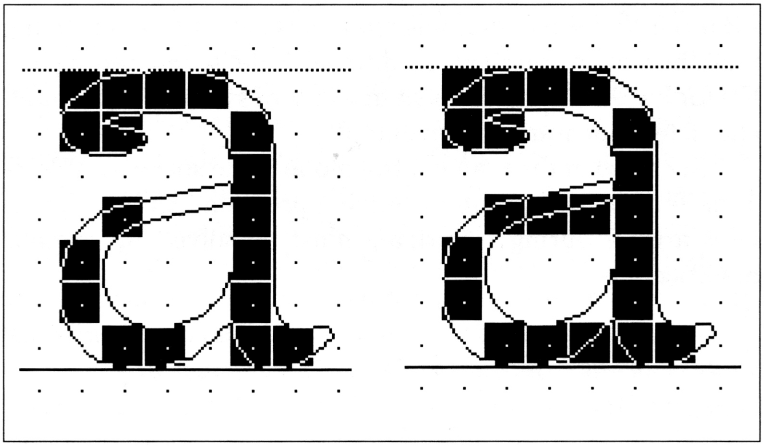


Fig. 140
The character «a», with and without dropout control.

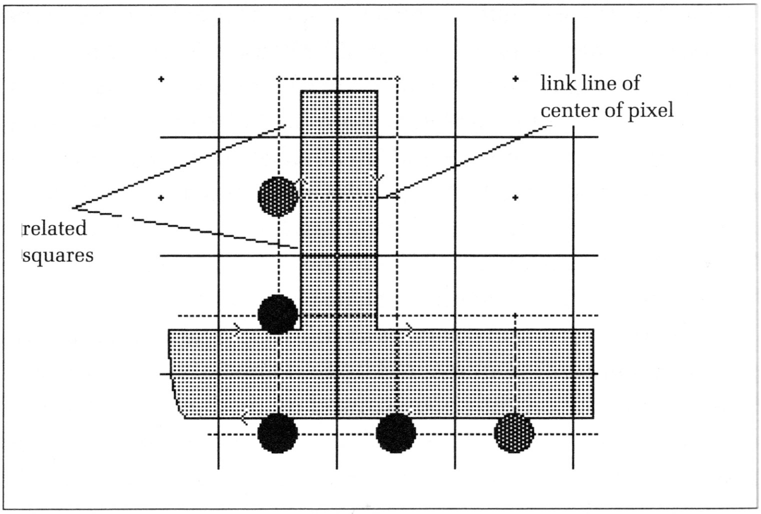


Fig. 141
To demonstrate scan control mode

In the above example, the simple rasterizing method would turn on no pixels at all. Black and gray filled points symbolize pixels to be turned on using the first variation of the scan control mode. The gray-filled pixels would not be turned on using the second variation, though, in actual practice, it would be adequate to use grid fitting.

The scan control mode should only be applied when dropout or gapping cannot be eliminated any other way. One reason for this is that this rasterizing method is more time-consuming than the simple approach. Additionally, deliberate setting of a left (or lower) pixel can result in a lack of symmetry in overall character appearance when output.

Such use of the scan control mode is the sole way a TrueType font can affect the rasterizing program. All other problems arising during rasterizing must be solved by contour modification.

8.3.5 The Instructions

Instructions incorporated into the font are read and processed by a program called *TrueType interpreter*. Using some key instructions as an example, this section provides a general overview of instruction structure and associated data.

Very much like a programming language, TrueType provides an extensive set of instructions for fitting characters to the required output raster.

Operator Code

The TrueType format operates by use of some 200 instructions. Each of these is identified by a number between 0 and 255, the so-called *operator code*. The TrueType interpreter identifies instructions solely by this number. For documentation purposes, however, it is useful to assign suitable names or abbreviations to them, which terms describe the function of each instruction. Operator code 101, for example, inverts (negates) the sign of a number, consequently termed *Negate*, or simply NEG.

Interpreter Stack

Most instructions operate with parameters. The NEG instruction, mentioned above, must be informed which number it has to negate, and must also be able to output the result. The TrueType interpreter has, for this purpose, a special memory area called the interpreter stack. Numerical values (parameters) are loaded into this stack and queued, to be retrieved in reverse order (LIFO structure – **L**ast **I**n, **F**irst **O**ut). The stack may be accessed only by instructions themselves. Parameters residing in a character's instruction block, are transferred to the stack using special statements found prior to actual instructions. The instruction code 176, for example, transfers the byte following it to the stack as a numerical value.

The byte sequence 176,32 thus transfers the value 32 to the stack. If this instruction is immediately followed by operator code 101, the NEG instruction mentioned earlier, the last element written to the stack will be read and removed from the stack. The NEG instruction then inverts the sign of the number, now becoming -32, and transfers it back to the stack. This element has now become the «last in» and can be read by the next instruction.

The entire purpose of these instructions is in fitting outlines to the output grid, which is effected by the moving of specific on-curve and off-curve control points. The TrueType format has a special set of instructions called move instructions, for these operations, to be dealt with next.

Freedom and projection vector

Distances are measured/compared in the direction of the projection vector. The actual shift is made in the direction of the freedom vector.

Common to all such instructions is the manner in which the move direction and distance are measured.

The direction in which points are moved is defined by a special direction vector, the freedom vector. The distance which points are moved is measured by still another vector, the projection vector. A point is moved along on the freedom vector until the shift corresponds to a specified distance as projected along the projection vector. The freedom and projection vectors, may thus never be at right angles to one another. A diagonal line joining two points in a character, for example, can be moved relative to a parallel reference line without affecting routing of other lines in the character (see Fig. 142).

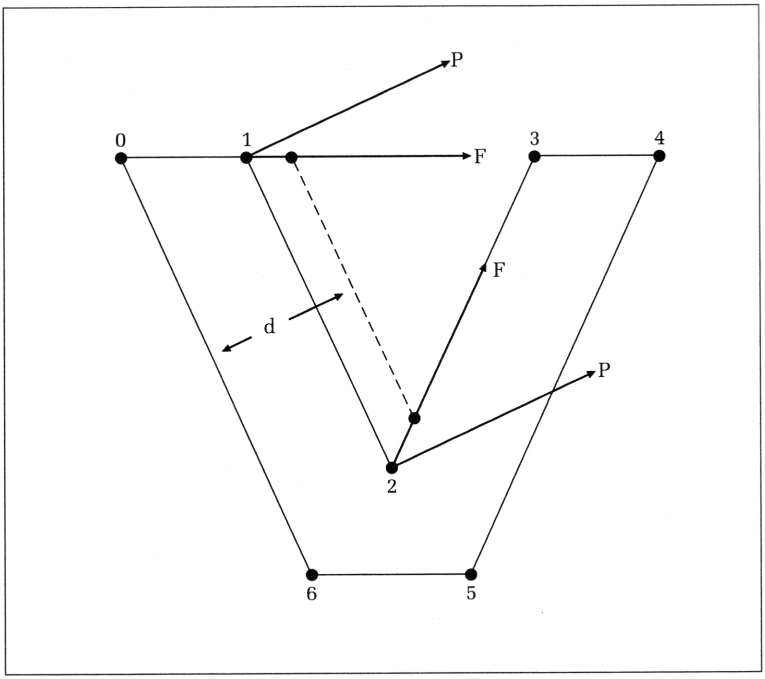


Fig. 142
Here, the freedom vector (F) and projection vector (P) have been set to check the width of a diagonal stem. Points 1 and 2 are moved, but the directions of the lines 0...1, 1...2 and 2...3 remain unaffected.

Graphics State

The graphics state, a memory area reserved by the interpreter, is another storage area used to store instruction parameters. These are value assigned defaults, generally remaining constant over an extended series of instructions. They need not be written to the stack each time they are required.

These graphics state variables – including freedom and projection vectors, for example – can only be modified by special instructions. The instruction SFVTL (**S**et **F**reedom **V**ector **T**o **L**ine – operator code 8) sets the freedom vector parallel to a line defined by two points specified in the stack.

The grid fit itself is performed by controlled shifting of on-curve and off-curve points with the help of move instructions.

Move Instructions

As with most instructions, the move distance is not given directly as a parameter. Were this the case, move distance would require a laborious computation. Each move instruction has a «target objective», and the actual move distance is always derived from that objective. Stem control on the lowercase «m» is used as an example demonstrating this in actual practice (see Fig. 139).

In the non-adjusted version of the raster image, all three stems in the «m» have varying widths. The first stem is two pixels wide, the other two are 1 pixel in width. The non-rasterized outline stem is a little wider than a single pixel. At some places on the character, the first stem, two pixel centers are located within the stem outline. Elsewhere (the second stem), a single pixel center lies within the stem boundaries. The stems are given non-uniform widths when that character is rasterized.

MDAP Instruction

The target objective of the MDAP (**M**ove **D**irect **A**bsolute **P**oint – operator code 47) instruction is to move a point on the freedom vector in such a way that the projected move on the projection vector is an integer multiple of the pixel distance. The left borderline of the left stem in figure 139 (all points on this line) are thus shifted in the x direction to the next pixel border. In this example, freedom and projection vectors both run in the x direction.

Round State

Absolute move instructions shift points irrespective of the movement of other points.

The MDAP or other instructions do not necessarily align coordinates to pixel borders. Adjustments (or rounding) is defined by a graphics state variable known as Round State. This variable enables pixels to be fitted either to lines mid-way between two grid squares or to pixel center points rather than pixel borders. A period, phase and threshold for the rounding process may also be specified, maximizing the function's flexibility (see Fig. 143).

MDRP Instruction

The right stem edge or border (all points along this edge) now must be moved so that its distance from the adjusted lefthand edge corresponds to the stemwidth of the original outline; the value of which is rounded off to the nearest pixel. In this case, this means that the right stem edge must be shifted exactly by one pixel. When the same process is applied to each stem, those stems identical in the original outline character will remain identical after the rasterizing process. A relative move of this type is executed by the MDRP instruction (**M**ove **D**irect **R**elative **P**oint, operator code 204).

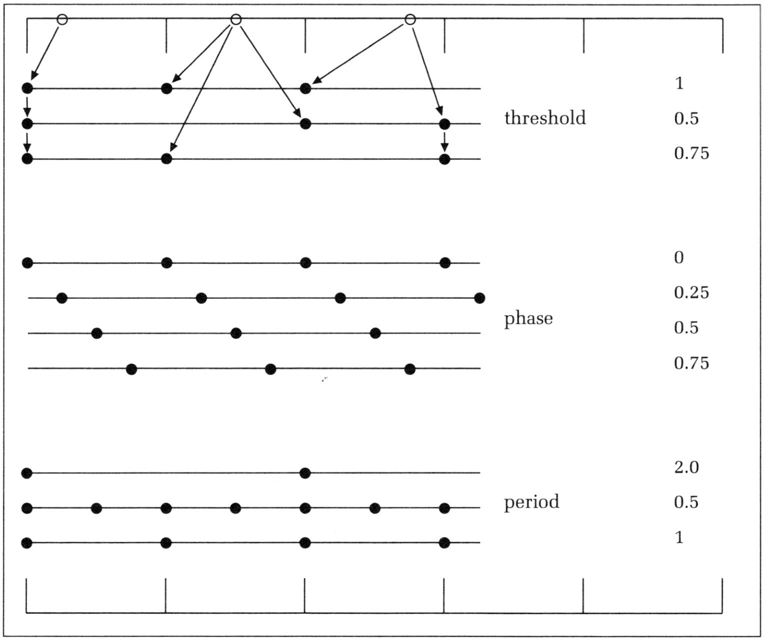


Fig. 143 Rounding positions with variations of the parameters period, phase and threshold in x direction. The period is the distance between the rounded values, measured in pixels. The phase is the distance between the rounded values and the integer multiple of the period. The threshold defines a position within the period; when this threshold position is reached, the converter goes to the next value.

Reference Points

The index of the point to be moved is entered in the stack, as it is with MDAP instructions. The point serving as a reference for a relative move, a point on the left stem edge in our example, is located in the graphics state variable rp0 (reference point zero). This variable can be set to the required index using a special instruction, or will be assigned automatically by a preceding move instruction in normal operation. The MDAP instruction automatically sets rp0 to the index of the point it is shifting.

Minimum Distance

Relative move instructions enable the distance between a point and a reference point to be rounded to a chosen value, e.g. an integer multiple of one unit of the target grid.

The MDRP instruction, in addition to other things, ensures that the distance to a relative point always remains as large or larger than the value specified in the mimimum distance graphics state variable. When *minimum distance* = 1, the stem always remains at least one pixel wide, making it impossible for the stem to be obliterated. One MDRP variant, however, does not adhere to the minimum distance.

Single Width

The method discussed to this point ensures identical-width stems in an original outline maintain an identical number of pixels in a bitmap format. Outline stems slightly varying in width may result in an extremely uneven appearance rendered in bitmap format, especially at lower resolution or smaller pointsizes. Variations of just one pixel can create a vast difference to the raster image where slight variations become much more pronounced. Stems with 1.5 pixels width, for instance, are made 2 pixels wide in bitmap format, those at 1.4, will be reduced to 1 pixel width. The MDRP instruction instruction can suppress this effect by setting all stems of characters smaller than a defined output size to a uniform width. This value is stored as the *single width value* in the graphics state.

Single Width Cut In

MDRP checks whether the relative distance (stem width in the original outline) differs from the single width value by less than the single width cut in value, another graphics state variable. When such is the case, the single width value is used rather than the original distance. Stem widths approximating this value (measured in fractions of pixels), are assigned identical values. This is, in fact, a general font control value, as all characters can work with the same graphics state variables. The cut in value is applied to all pointsizes small enough to warrant this method of stem control.

Direct move instructions essentially work with the distances and coordinates of the original contour.

MIRP Instruction

Single width control is a suitable method for maintaining uniform distances at either low resolution or small pointsizes, where checked distances do not exceed one or two pixels on the raster image. For larger size fonts, however, font-specific width and length variations may again be represented by varying the number of pixels used to display character elements. A more feasible approach with larger pointsize fonts is to group stems (or other distances) into similar width categories, each assigned a specific value. When this is the case, the MIRP instruction (**M**ove **I**ndirect **R**elative **P**oint, operator code 236) is used rather than the MDRP instruction.

Control Value Cut In

MIRP checks whether the relative distance (stem width of original outline) differs from that value defined in its control value table parametrized (using its index the stack) by less than the graphics state variable control value cut in value (measured in fractions of pixels). When the stem width differs by less than this cut in value, the control value is used instead of the original width. Other than that, the approach is the same as with MDRP instructions. For fonts output smaller than a certain size, any widths close to the control value are automatically assigned identical numbers of pixels. The single width control remains in effect, the selected width (original or control value) is treated as the original width in the MDRP instruction, possibly substituted by the single width value.

Distance Control

Variations between distances at higher resolutions do not undergo these controls, and are output as originally intended in the font's design. As resolution and pointsize decrease the more control values are utilized. At the smallest output size, all stems are simply set to a single width value. These distances, incidentally, are not limited to stem widths, but apply also to the width and height of serifs or shallow curves which are monitored in precisely the same way (flex control).

Twilight Points

It is sometimes necessary to check distances not measurable relative to a reference point. For instance, cap height overhang can be checked even when the character contains no points on the cap height line. Such checks are managed using support points, so-called twilight points. The same as points in the character itself, twilight points can move around and be used as reference points. The max profile table defines the quantity of twilight points available.

Indirect move instructions access the coordinates and distance values in the control value table. They enable typographic values such as stem widths and positions, also x-heights and cap heights, to be uniform for the entire font.

MIAP Instruction

Our picture would be incomplete without a brief mention of the MIAP (**M**ove **I**ndirect **A**bsolute **P**oint, operator code 63) instruction. It operates similarly to the MDAP instruction, but can take a value from the control value table rather than using original coordinates. Figures 144 and 145 summarize functions and effects of the four move instructions.

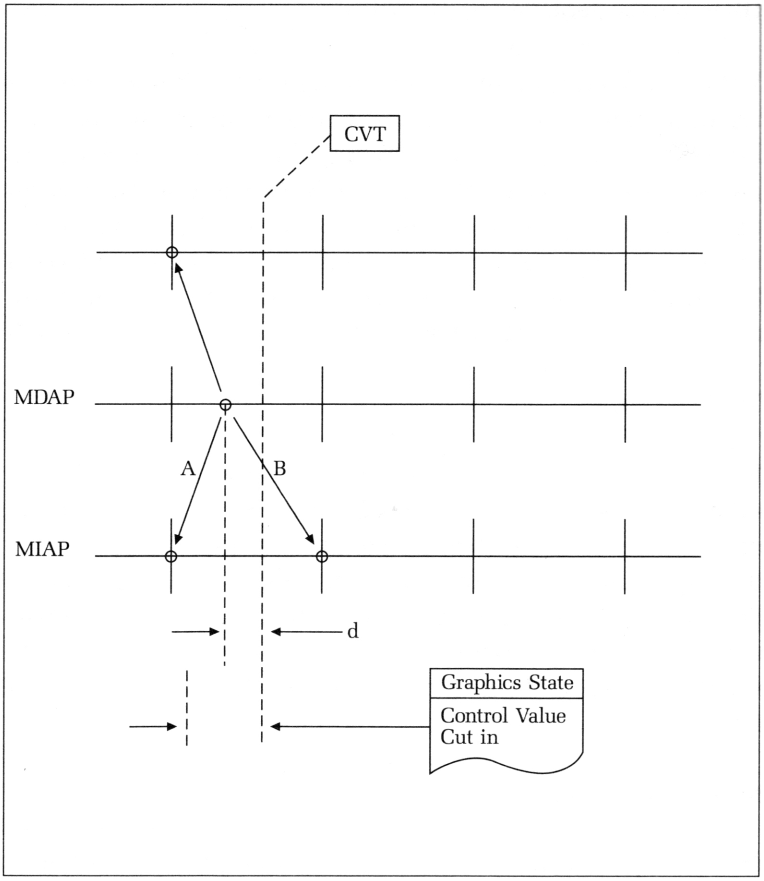

CVT
MDAP
A
B
MIAP
d
Graphics State
Control Value Cut in

Fig. 144
Absolute move instructions (freedom and projection vectors in x direction)

MDAP (=**M**ove **D**irect **A**bsolute **P**oint) fits/adjusts the original coordinate (case A).
MIAP (=**M**ove **I**ndirect **A**bsolute **P**oint) takes a value from the control value table (cvt) when the distance between this value and the original coordinate, measured in pixels, is smaller than the control value cut in value from the graphics state (case B).

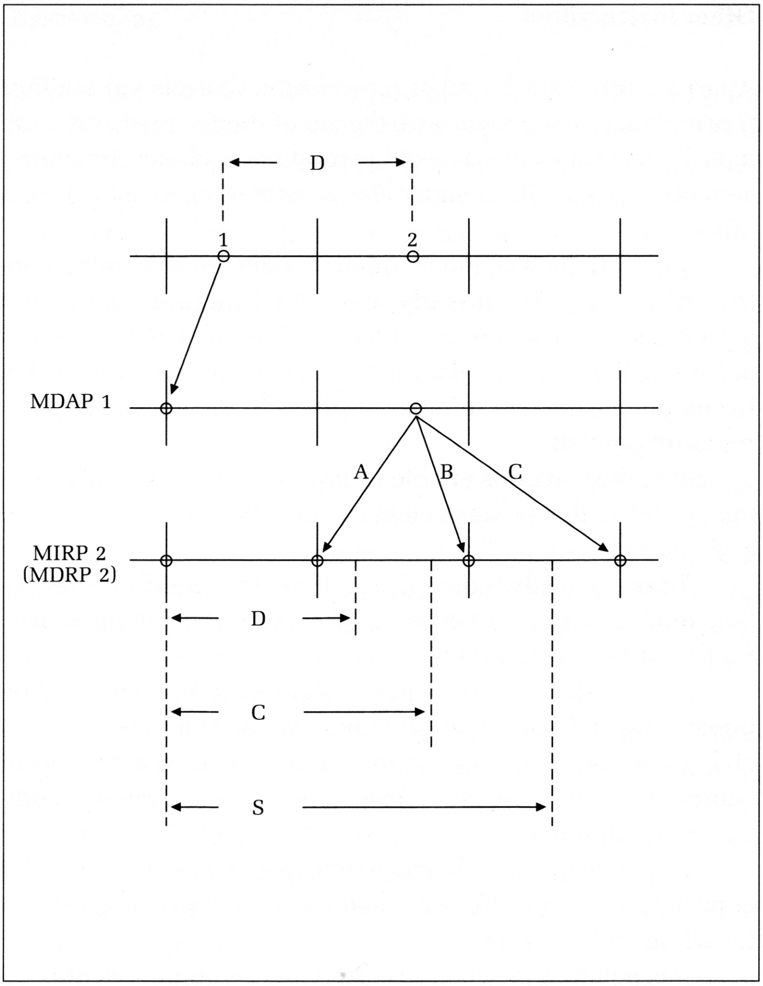


Fig. 145
Relative
move instructions
(freedom and
projection vector
in x direction)

MDRP (=**M**ove **D**irect **R**elative **P**oint) sets the distance to a reference point (adjusted with MDAP, for example) to the (fitted) original distance (D) (case A).

MIRP (=**M**ove **I**ndirect **R**elative **P**oint) takes a control value (C) from the control value table when this value, measured in pixels, deviates from the original distance by less then the control value cut in value (case B).

Both of these instructions will access the single width value (S) when the selected distance (original or control value) deviates from the single width value by less than the single width cut in value, measured in pixels (case C).

Other Instructions

Apart from the stack and graphics state, there is yet another free memory area which, with the aid of special read and write instructions, can store data of the producer's choice, including general font data. Its memory size is defined in the max profile table.

The processing sequence of instructions can be influenced by conditional execution of instruction blocks. Sub-programs can also be integrated.

There are instructions to read or write entries in the control value table. We already mentioned these to access the graphics state variables and others which, in turn, are used to access further internal data (point coordinates, distances between points, current number of pixels in the em square, current font pointsize).

Some instructions enable interpreter stack contents to be manipulated, that is, stack elements may be re-ordered, copied and deleted.

There are arithmetical instructions for addition, subtraction, multiplication and division of numerical values as well as for rounding functions, etc.

The instruction processing sequence can be controlled by integrating IF-ELSE conditions into instruction blocks, or by giving absolute jump instructions to be executed after logical comparisons. It is also possible to define sub-programs or additional instructions.

Output demands of black- or white-writing devices may be adjusted by using diffferent versions of the move instructions for white and black areas.

Interpolation instructions, used to interpolate unadjusted points between two points already having undergone repositioning, are of some significance ensuring outlines retain their overall balance.

The *adjust angle* (AA) instruction is used to modify the slope of diagonals in a way that aesthetically pleasing periods are obtained during rasterization. Another remarkable group are delta instructions, which, with certain pointsizes (in pixel units) can be used to move points specific distances or adapt control values. The objective is to turn specific pixels in the raster image either off or on.

Assessment

All instructions affecting a character outline operate on a purely geometrical basis, typographic terms (stem, serif, etc.) not used at all. Font producers have the responsibility of deciding which type-specific features the scan converter should recognize and how these features are to be checked and monitored during rasterizing. The price for this high degree of flexibility is considerable added effort required for production.

8.4 F3 Unix

by Jochen Lau

The fourth standard for «intelligent outline fonts» outlined is the F3 language from SUN Microsystems. F3 is a general programming language used specially for description of geometric (character) forms and their conversions to ensure optimum bitmap-format output at various resolutions.

F3 technology was originally developed by Folio, an enterprise founded by former employees of the Imagen company. The format's name also comes from this era: Three «Fs» for **F**olio **F**ont **F**ile format. As the product began to make some headway on the market, SUN Microsystems took it over, and, in consequence, F3 technology is (or was) an integral part of various SUN products:

F3, a Sun product

Open Windows (graphical user interface), NeWSprint (printer technology) and TypeScaler, a program to convert F3 format fonts into bitmaps for various output devices. As such, it is comparable to Adobe's Type Manager (ATM) or NIMBUS from URW.

Fonts are coded as F3 to overcome problems inevitably arising as outline fonts are rasterized. The F3 interpreter fits the font to the target grid, ensuring that identical elements, such as same-width stems, do, in fact, remain the same. This modified outline description is then converted to bitmap format by the F3 rasterizer, which also checks design integrity, making certain that no dropouts are created.

Two level process, interpreter + rasterizer

Character Contours

Contours of F3 type characters are described as closed curves using straights, cubic Bézier curves or conics. Unlike other described technologies, F3 allows two different mathematical methods to be used to describe curved lines in the same character.

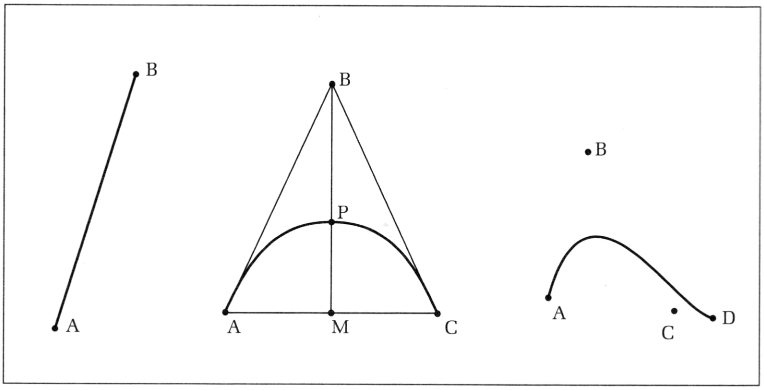


Fig. 146
Examples:
a straight line,
conic and
Bézier curve

As in other systems, it is assumed that inner and outer contours run in opposite directions, not overlapping. Only then will the rasterizer fill outlined areas correctly. Generally, the outer contour runs in a clockwise direction, and so on.

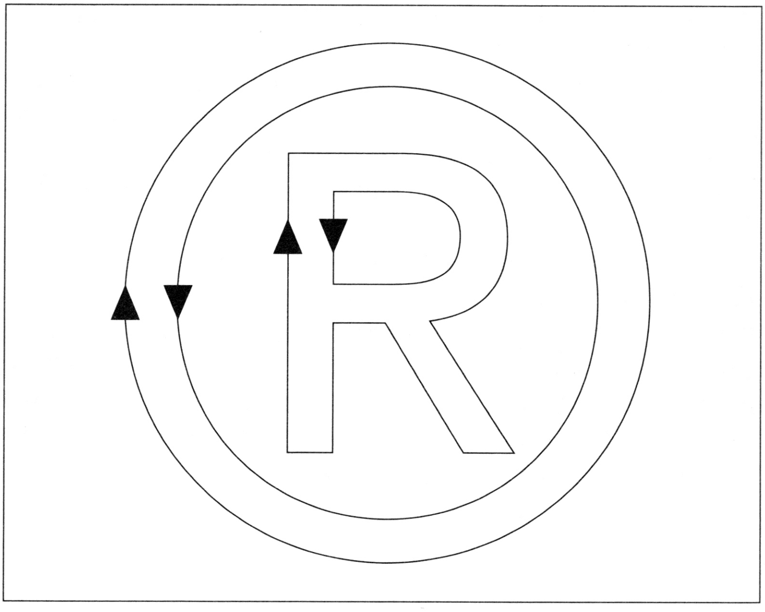


Fig. 147
Outer and inner
contours must run
in opposite
directions

F3 Programs

PostScript Type1 and NIMBUS typefaces contain information to identify and describe character elements (such as stems) requiring special treatment, while these typographic terms play less of a role in the F3 format, even less than in TrueType format. U.S. typeface producers frequently use the term «font program», probably coming the closest to describing what is involved in the F3 format.

A character is an F3-program.

A program in the F3 language is used to describe a kind of «master outline» for each character in the font. Apart from the points describing contours, the program also contains instructions how points are converted when coordinates in the target grid are generated during the gridfit. This target outline is described in a system whose axes run parallel to lines and columns of the target grid and whose units correspond to the pixel size of the target. Other than actual outline data, the target outline contains no other information. To perform character calculation, the program must access a transformation matrix which forges a link between character and grid coordinates. This transformation matrix is the sole input parameter received by the interpreter.

F3 permits any algorithms.

No rules dictate structure of an F3 program. In the simplest case, it would simply execute coordinate transformation, not dealing with any of the usual rasterizing problems. Due to its general nature, the language allows use of any algorithm to adapt the outline to its target grid. For instance, calculations NIMBUS conducts implicitly when interpreting stem instructions could be integrated into an F3 typeface as part of a program. As in TrueType, F3 fonts contain all the «intelligence» needed as a built-in component.

Results on the stack

The result of an F3 program is delivery of fields to the stack. These fields, describing the grid-adapted outline, are: the auto advance vector (total character width), origin displacement vector, the bounding box, and, for each contour, the curve segment type and contour points.

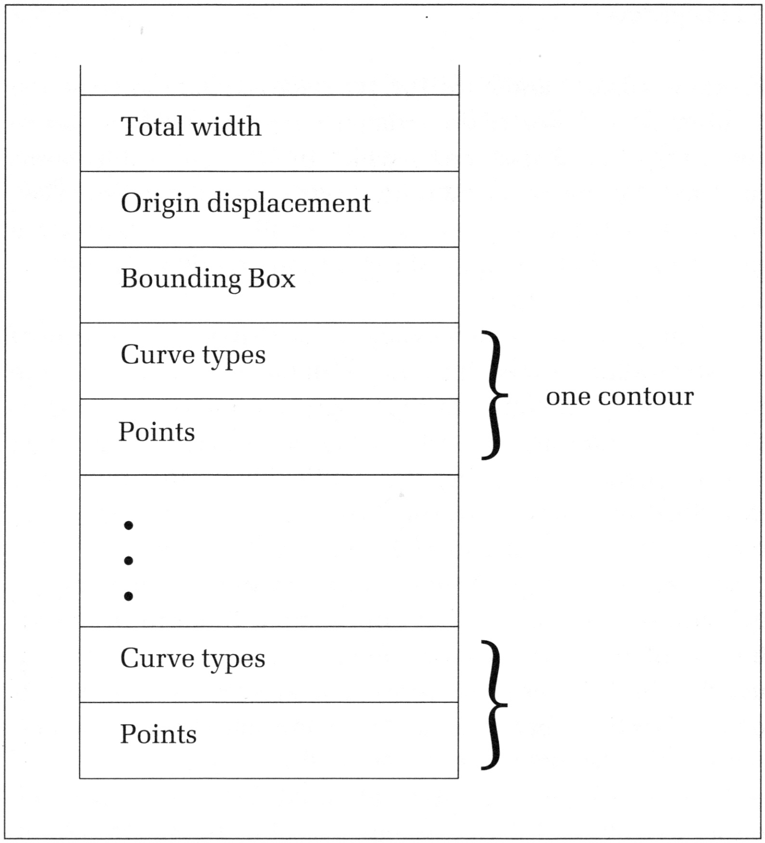

Total width
Origin displacement
Bounding Box
Curve types
Points
one contour
Curve types
Points

Fig. 148
Results of the interpreter run

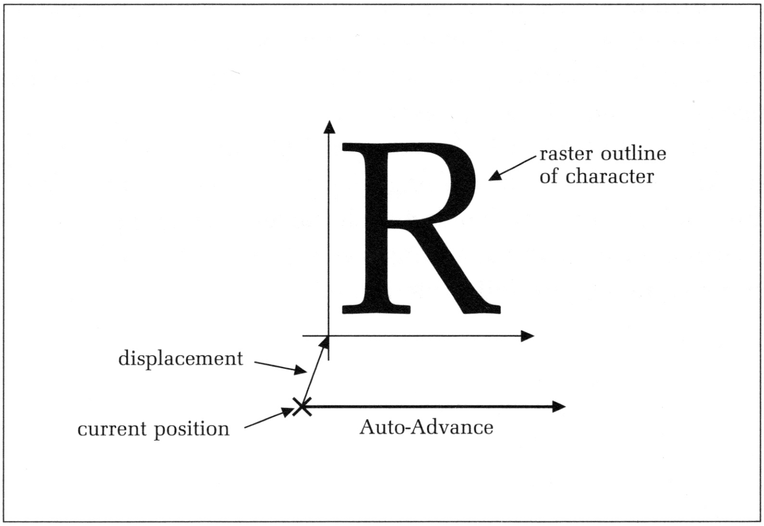

raster outline of character
displacement
current position
Auto-Advance

Fig. 149
Results in diagram form

F3 Language Elements

Statements of F3-coded typefaces are stored as a sequence of binary bytes. Generally, language elements are specified by an ID in the first byte and subsequent bytes, containing parameters.

Numbers are given in various formats, as short and long 16-bit integers or as 2- and 4-byte floating point numbers; it is also possible to define fields of these types. In addition, procedures may be assembled from the following simple operators:

- Operators to edit fields (create, determine size of, etc.)
- Assignment operators (to assign values to memory cells and fetch values from cells)
- Stack operators (pop, push, etc.)
- Numerical and logical operators (+, – and sin, cos, etc.)
- Control operators (conditions, loops, etc.)
- Graphics operators (such as TRANSFORM: applies the transformation matrix to one or more points, PIXRATIO: states the device-specific ratio between an ideal and a real pixel on the device, thus enhancing quality of results).

Please consult (Lit. SU 91) for a more detailed description on how these operators work and with which values they are coded.

F3 Font File Format

As a rule, an F3 font consists of a font header, an index table for character access, a preamble and at least one character.

The key entries in the font header are addresses of other font components, plus the font bounding box.

Preamble to compute global data

The preamble and characters are stored as F3 programs (see above). When a character is rasterized, the preamble is always executed first, providing data and operators valid for all characters. The F3 font files contain simply the outline descriptions, scaling statements and total character widths. As application programs need further information in order to access F3 fonts, this information is stored in separate files, an absolute necessity in an F3 font.

Character numbers, no names

Unlike PostScript, F3-format characters are accessed via a unique number, used as an index for the access table. The F3 Map File lists all characters contained in a font, plus the Post-

Script names to which they are assigned.

Metrics in the AFM file

A separate file exists for spacing values of left and right side bearings, and kerning, for instance. F3 does not use its own format for this information, preferring the Adobe Font Metric (AFM) instead. Use of the AFM structure means that F3 Map Files must be accessible as characters are addressed under their numbers in the F3 font, and their names in the AFM file.

Conclusions

Twenty-eight years ago, the CRT machine was invented by Dr. Hell, and at URW we have been dealing with digital formats for twenty years. We consider ourselves fortunate that we began our work utilizing the control of drawing machines to cut original letters for photosetting, i.e. from the start we had to concentrate on quality of digital type reproduction due to the size and accuracy of output required. Most typesetting machines delivered maximum letter heights of approximately 3.1 cm. Drawing machines use 10 cm as a standard height, with large images 100 cm and larger.

These circumstances led us to the best database for the purpose. We held our peace for a considerable time before proclaiming the IKARUS format as the only correct one for this purpose since, at first, we could not believe that storing hand-digitized data was the most sensible procedure. We found it to be too simple because, after all, the basic concept did not require a great deal of thought.

The IKARUS format contains no math.

It is likely that one dealing with graphics eventually ends up differentiating between three types of points (corners, tangents and curves), also a very natural process when analyzing letter outlines. A fourth type of point is the Starting Point of importance to us only in enabling the differentiation of individual closed outlines (for example the letter «O» has two outlines).

The coding of IKARUS points is, consequently, extremely simple. X and Y coordinates are coded with one word each and the point type is put into the sign bits of these two words, i.e. 4 bytes per point (see Appendix H). Discussions with critical users provided a further argument for the worthiness of the IKARUS format: hand-digitizing, by its nature, usually leads to a minimum of points being designated, as people tend to favor saving work. Thus the IKARUS method is that requiring the least number of contour points to reproduce an original letter, regardless of how it was drawn. In this respect, it is of no consequence which method is used to produce smooth and rounded connections of hand-digitized points. We still maintain that in addition to our method (IK interpolation), spline interpolations and Bézier curves are of some use. The IK interpolation method was discussed in Chapter 6.3 with an explanation of why we continue to use our method.

Considerations on this subject led to a more detailed investigation of the properties of other formats.

9.1 Properties of formats

The following summary lists format properties in the sequence in which, during their short history, technicians have become interested in them.

E0: **Electronics (speed)**
Cost of electronics for the generation of control signals.

E1: **Storage capacity (high res)**
Storage capacity for high resolutions.

E2: **Storage capacity (low res)**
Storage capacity for low resolutions.

E3: **Sizing**
Possibilities for creating different type sizes from one original.

E4: **Transformations**
Possibility of displacing, rotating, italicizing and deforming type faces.

To these already well known and accepted properties we have added a few of our own:

E5: **Input**
Keeping input as simple as possible.

E6: **User friendliness**
Simple data structure for easy corrections.

E7: **Presentation**
Accurate output on flat bed drawing machines to allow

comparisons with the original, and the output of lettering on self-adhesive material.

E8: Soft-scanning
Accurate, closed mathematical curve description in order to be able to use calculation methods to arrive at other digital formats, e.g., bitmaps, vector formats and run length codes.

E9: Programmer friendliness
Cost-effective programming of modifications; easy programming for new machine formats.

In the following table we have evaluated the properties of the individual formats as follows:

15 = very good
8 = satisfactory
0 = unsatisfactory.

The groups of properties have been divided into subtotals for

• machine format	(E0...E2)
• typesetting machine format	(E0...E4)
• data base	(E5...E9).

Particular attention has been paid to properties E0 and E1 in the past, and we shall pay attention to these also in sections 9.1.1 and 9.1.2.

Fig. 150 Comparison of formats

	Machine-format			Type-setter-format		Data base format					Sums				
	E0	E1	E2	E3	E4	E5	E6	E7	E8	E9	E0+ E1	E0+ E2	E0- E4	E5- E9	E0- E9
Ikarus-format	1	13	13	15	15	10	15	15	15	12	14	14	57	67	124
DI-format	5	8	8	15	12	15[3]	12	15	8	10	13	13	48	60	108
VE-format	7	15[1]	12	12	15	15[3]	10	12	7	8	22	19	61	52	113
SC-format	9	15[1]	4	3	5	15[3]	7	8	6	2	24	13	36	38	74
BI-format	9	15[1]	0	3	5	15[3]	6	6	5	2	24	9	32	34	66
GS-format	9	13	0	3	5	15[3]	3	4	2	2	22	9	30	26	56
Adobe-format	5	13	13	15	10	7	9	13	8	10	18	18	56	47	103
Bitstream-form.	5	8	8	12[2]	12[2]	10	15	15	15	12	13	13	45	67	112
Coueignoux-f.	9	15	15	10[2]	10[2]	10	10	6	8	8	24	24	59	42	101
Metafont-format	0	3	3	12[2]	10[2]	3	12	10	8	15	3	3	28	48	76
Purdy-format	10	13	13	12[2]	10[2]	4	10	8	8	8	23	23	58	38	96
Xerox-Spline-f.	5	10	10	12[2]	10[2]	12	9	9	8	8	15	15	47	46	93
open. Vector form.	13	15[1]	8	3	5	12	8	10	6	2	28	21	44	38	82
closed Vector format	7	15[1]	8	7	10	12	10	12	7	8	22	15	47	49	96
Run length format	14	15[1]	4	3	5	12	7	8	6	2	29	18	41	35	76
Bitmap format	15	15[1]	0	3	5	12	6	6	5	2	30	15	38	31	67

[1] Look also table and histograms for storage needs for different rasters.
[2] These formats have a primary resolution of less than 4000 × 4000 pixel per em.
[3] These formats could be calculated by computer.

E0 = Electronics (speed)
E1 = Storage capacity (high res.)
E2 = Storage capacity (low res.)
E3 = Sizing
E4 = Transformations)
E5 = Input
E6.= User friendliness
E7 = Output
E8 = Soft-scanning
E9 = Programmer friendliness

9.1.1 Electronic requirements for the generation of control signals

It is evident that electronic effort required increases proportionately as digital reproduction of used formats becomes more complex or compressed. We will attempt to assess manufacturing costs of appropriate electronics – not developmental costs – and summarize these below. The table contains details in the form of «number of IC boards» required for generation of control signals from a particular digital form of data storage. The «number of IC boards» is based on the electronic status quo of the year 1985. The more time passes by, the smaller the numbers become, but the relative proportions remain.

	Electronic 1	Electronic 2	Electronic 3	Electronic 4
IKARUS-Format	-	-	-	15
DI-Format	-	-	8	-
open vector format	3	3	-	-
closed vektor format	4	4	7	10
Running length f.	2	2	-	-
Bitmap format	1	1	-	-
Adobe-format	3	6	10	10
Bitstream format	-	-	-	10
Coueignoux-format	6	6	6	6
Metafont-format	-	-	-	-
Purdy-format	3	5	5	5
XEROX-Spline-form.	-	-	10	10

Electronic 1 up to 4 perform all about 500 characters per second in 10 pt size.

Electronic 1: low resolution (VDT's, laser printer)

Electronic 2: hight resolution (laser recorder, CRT-machines)

Electronic 3: high resolution with performances E2 and E3 (high resolution and sizing)

Electronic 4: high resolution with performances E2, E3 and E4 (high resolution, sizing, transformation).

Fig. 151
Cost of electronics on basis of number of IC boards in double Europe format.

Fewer IC boards for the more complex formats would be required if one used the following method to generate the control signals:

- Stores type as a basic data format, in itself complex.
- Prior or parallel to setting the text, typeface commands are searched for font change commands.
- A preparatory process is initiated on a standard IC board (e.g., a CPU with 512KByte RAM memory), a complex format for the required type is converted relatively slowly (approximately 40 char./sec.) to a simple machine format by a program.
- The simple machine format is then stored in a buffer prior to actual typesetting.

9.1.2 Storage requirements

The framework of the IKARUS concept makes it relatively easy to simulate special encodings, and thus finding the storage requirements for corresponding formats. Following we show a table and two graphic presentations. The table contains an estimate of the storage required for complex formats. Diagrams show curves of the storage required as a function of the raster per em for complex as well as simple formats.

Storage requirement for complex formats is virtually independent of resolution.

	Bytes per digitized point or element	Points per character	Bytes per character	General information and data for a typeface	KBytes per 100 Latin-characters	KBytes per 5000 Kanji-characters.
IKARUS-format	4	50	200	2	22	4950
DI-format	8	75	600	2	62	13950
VC-format	3	75	225	2	24,5	5513
Adobe format	10	20	200	2	22	4950
Bitstream-format	7	75	525	4	56,5	12713
Coueignoux-format			25	10	12,5	2813
Metafont format			400	20	60	13500
Purdy-format	5	40	200	5	25	5625
Xerox-Spline-format	20	20	400	2	42	9450

Fig. 152
Storage requirement for different formats

The numbers show net values. Optionally, one can add storage for:

- width tables,
- kerning/kissing tables,
- keyboard layouts,
- special typeface parameters and indicators.

Storage requirements of **Kanji** characters can be estimated on the basis of IKARUS digitizations. These characters average 225 digitizations instead of 50, **thus requiring more than 4.5 times more storage space per character.**

Fig. 153 + 155 Storage requirement for high and low resolutions.

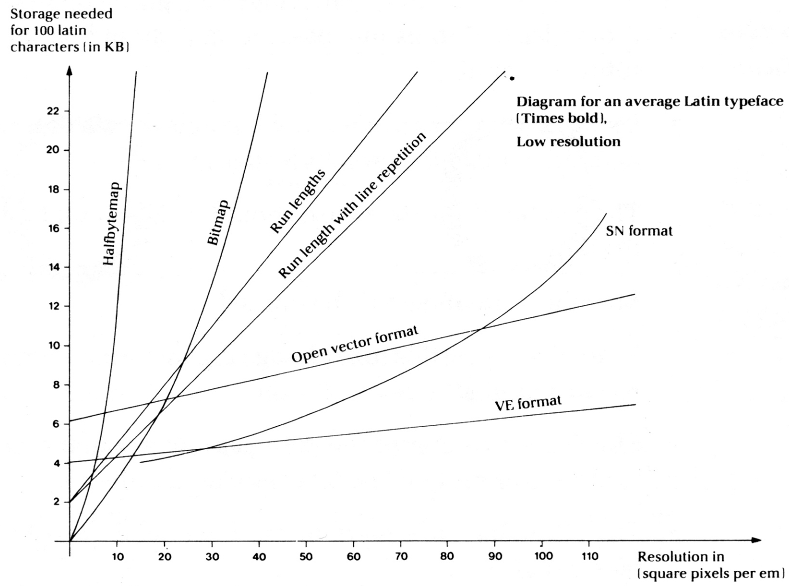


Fig. 153 Diagram for an average Latin typeface (Times bold), low resolution

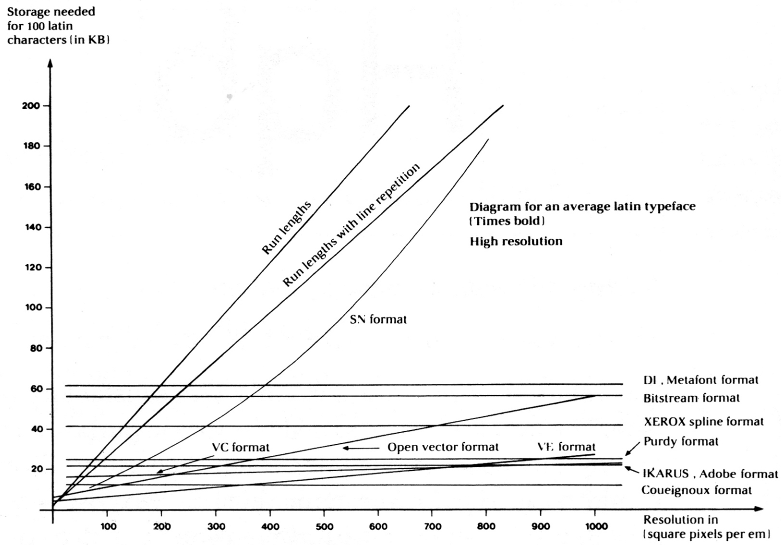


Fig. 154 Diagram for an average Latin typeface (Times bold), high resolution

9.2 Requirements to be met by formats

Here is a summary of our findings as to requirements to be met by formats:

- The pixel aspect ratio should be a square.

Typefaces should be stored in a database.

- A data base of fonts in a basic format should be set up for future machines.
- Using this data base, one should be able to calculate current and future machine formats (soft-scanning).
- The format of the data base should be based on hand-digitization.
- Machine formats should be unified.
- It should be possible to exchange digital formats among different manufacturers.
- The correct format of the data base allows manufacturers to change or expand the data more easily.
- The following em square division would appear to be the most sensible:

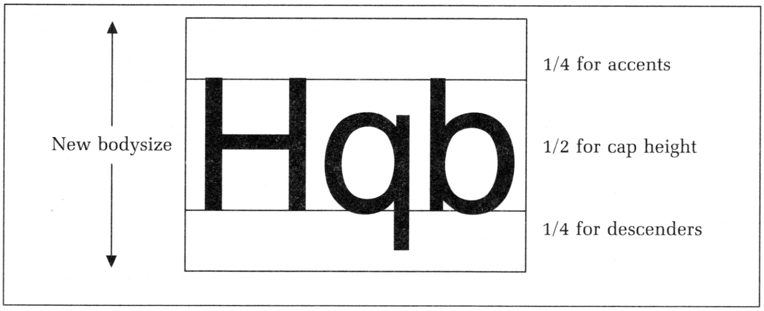


Fig. 155
We ought to dispense with the em of metal composition.

This produces the following:

new body size = old body size × 1.33 = 2.0 × cap height.

Several firms have introduced this with satisfactory results to date. The new body size is used for storage purposes not determining line spacing (leading) when typesetting. Even better would be to replace the term «em» by «exposing window», selecting this window with dimensions equal to or slightly greater than the new em.

9.3 Recommendations for manufacturers

- When the raster pattern is smaller than 50 by 50 pixels, representation of overhangs to correctly display round forms in an alphabet will have a detrimental effect, with curves appearing flatter than they should.

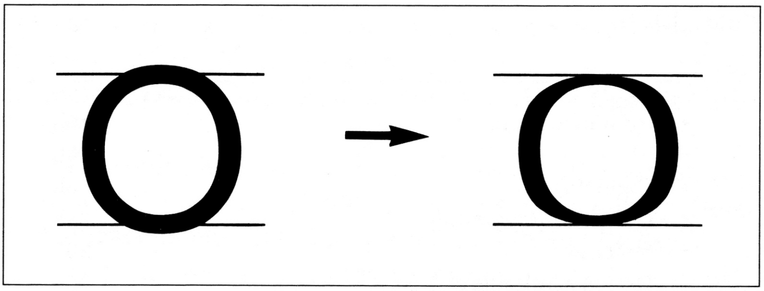


Fig. 156
Without overhangs, rounded character elements appear too flat.

- Laser printers should write with at least 600 lines/inch.
- Technicians should keep in mind that the development of a new typesetting machine requires approximately 5 to 10 persons for at least 2 years. If one already has a good data base, during this time 2 persons may work concurrently preparing type in the new machine formats.

In our experience to date, type is only developed after a machine is completed, and then as rapidly as possible. This practice must change!

- Descender length should be equal to ascenders to achieve better proportion.
- We would argue for fixed accents in a type library (data base).

Long kerning!
No compromises.

- Also we argue for complete and comprehensive kerning and touching tables in the data base.
- Efforts should be made allowing variable type and/or keyboard setups. Every data base should be flexible in this way.
- Type should be down-loadable from a host computer, not a fixed part of the typesetting machine.

9.4 Time required for producing type

In our type studio we digitized approximately 3,000 type versions to the end of 1992, and can speak from that experience. On average, 100 man-hours were required to digitize a font of 186 Latin characters (normal URW typeface layout) by hand in the IK format. For these 186 characters the work involved is divided as follows:

• Administration	10h
• Marking	8h
• Digitizing	8h
• Correction/operating	24h
• Spacing	20h
• Design of missing characters	30h
	100h

The last item is due to the fact that, on average, original drawings usually contain about 120 characters. The missing 66 characters are drawn, partly by hand, partly calculated using programs, or with the aid of graphic editors on a visual display terminal. For the hand-digitization of Kanji type characters in Hamburg, Shanghai and Tokyo, we have come to the following values by experience (this also applies to a 186 Kanji character font):

• Administration	10h
• Marking	24h
• Digitizing	24h
• Correction/operating	35h
• Spacing	–
• Design of missing characters	??h
	93h

With regard to marking and digitizing, Kanji characters have on average 4.5 times the number of points contained in the average Latin letter. Despite this, the data preparation and digitizing takes only three times longer, approximately 15 minutes per Kanji character.

With regard to corrections, it should be noted that Kanji characters are drawn in the middle of the em square and appear more individual than Latin letters. The vertical strokes of Latin type are usually uniform within all the caps or lower case letters of a particular typeface. This can only be achieved by a relatively large amount of corrective work. The originals are drawn by hand and therefore naturally have the inaccuracies of hand work, e.g., with respect to stroke widths. Fortunately, the same rigidity of rule does not apply to the individual Kanji characters, i.e. the amount of editorial work for Kanji is fairly limited.

We would, however, indicate the very large extent of a Kanji font, i.e. around 10,000 characters for «mincho» (text) and approximately 6000 characters for «Gothic» (display).

10,000 characters per Kanji font

The Japanese have – just as the Europeans or Americans – developed a large number of different type fonts and versions. We estimate that for all existing Kanji type one would require approximately 30 times the digitization work needed for all the Latin type in the western world.

In addition to the amount of effort involved in the preparation of IK formats, we also have an appreciable amount of manual work with the preparation of SC formats. The demands made by different firms and therefore the corresponding amount of work are very different.

The effort involved with soft-scanning is fixed:

First point size parameterization	10 hours
Other point size parameterization	4 hours

The next working steps take place on the graphic display. At this point we should perhaps say that some manufacturers of type offer digitizations which are of a quality lower than that of the results produced by automatic soft-scanning.

In our opinion corrections are always to be expected. There are just too many possibilities for graphic variations at joints, serifs and stroke intersections. No programmed procedure can be expected to deal with all these eventualities. On the other hand there are already high resolution machines available (60 lines/mm up to 100 lines/mm) which no longer require the correction of rasters made with soft-scanning.

The average amount of time required by us for the correction of 186 Latin type characters for each pointsize in a range of 50 × 50 up to 400 × 400 is about 15 hours, while the first pointsize of all pointsizes of a particular version usually takes between 30 to 45 hours.

The production of digital type is expensive.

The production of digital type is therefore expensive. It is for this very reason that when preparing a database, **the digitization of the typeface should only be required once in the life of that typeface**. Once it is stored one only needs a relatively small amount of rastering work for each machine, whether available now or yet to be developed.

Appendices

Appendix A

Type classification

While there are several competing methods, we follow DIN-classification based on proposals made by the ATYPI (Association Typographique Internationale). American and English standards use 9 groups of typefaces. We have added two others, (10) for black letters and (11) for eccentric typefaces. Group 11 is used to categorize all typefaces, which are difficult to read, and therefore of limited use, except perhaps in display advertising.

A few years ago, an ISO group was established to work on international font standards (ISO/TC97/SC18). Their work is completed and shows promising results (ref. ISO, AFII).

Venetian Serif

Characteristic of this group are the sloped cross stem of the lower case e and the sloped (45 degree) up-stroke of the stems of lower case characters. The axis of bowls is slightly slanted to the left.

This classification complies with DIN standards.

Centaur
Schneidler
Trump Mediaeval

Fig. 157 Examples of Venetian serif

Old Face/Old Style

The cross stem of the lower case e is horizontal. The terminals of the upper case and the vertical stems of the lower case curve smoothly into the serifs.

Bembo
Palatino
SABON-ANTIQUA

Fig. 158
Examples of Old face

Transitional

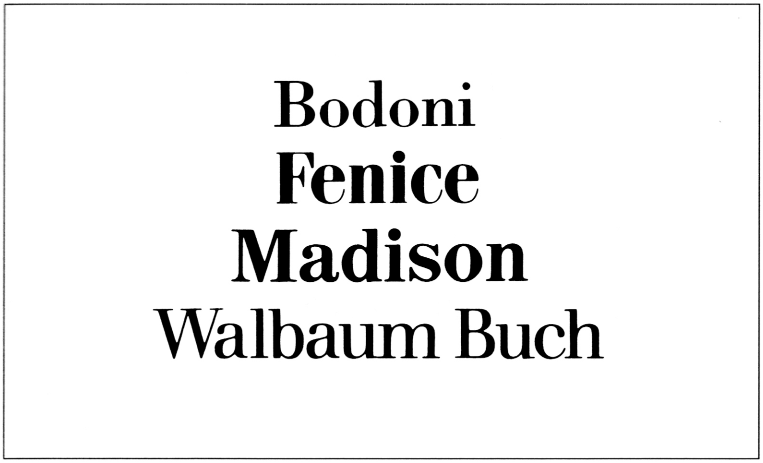
The differences between thickness of hairlines and stems are greater. The axis of the bowls is nearly vertical.

Baskerville
Caslon
Concorde
Times

Fig. 159
Examples of Transitional

Modern Face/Modern Roman

The serifs are smooth and horizontal. The stems do not curve smoothly into the serifs. The axis of bowls is vertical. Differences in thickness between hairlines and stems are more extreme here.



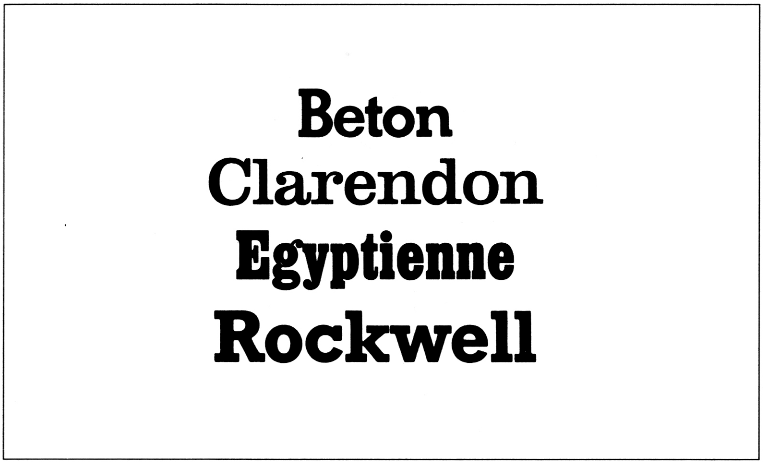
Bodoni
Fenice
Madison
Walbaum Buch

Fig. 160
Examples of
Modern face

Egyptian Slab/Serif Antique

Characteristic to this group are the slab serifs. Hairlines and strokes are mostly as thick as the terminals.

This is where technology first make its marks.



Beton
Clarendon
Egyptienne
Rockwell

Fig. 161
Examples of
Egyptian slab

Sans Serif/Gothic

This group of typefaces has no serifs, and is also recognized by the name «Grotesque». Crossbars and stems are optically of the same thickness.

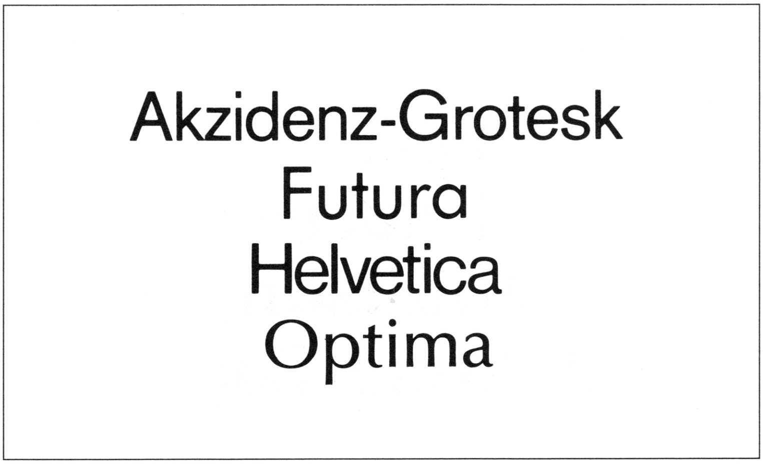

Akzidenz-Grotesk
Futura
Helvetica
Optima

Fig. 162
Examples of
Sans serif

Display Roman

These roman versions are not classifiable within the first six groups, due to their widely differing ductus.

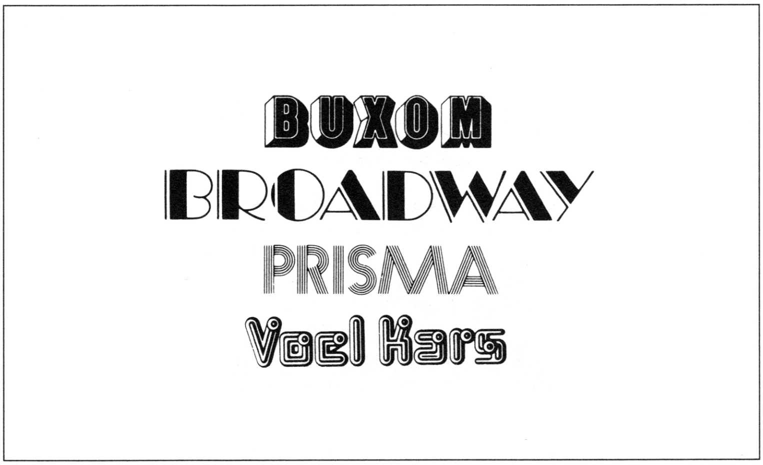

BUXOM
BROADWAY
PRISMA
Vogel Kars

Fig. 163
Examples of
Display roman

Scripts

These typefaces have been developed from writing at schools and chancelleries. The Roman italics are however not to be mistaken for members of this group.

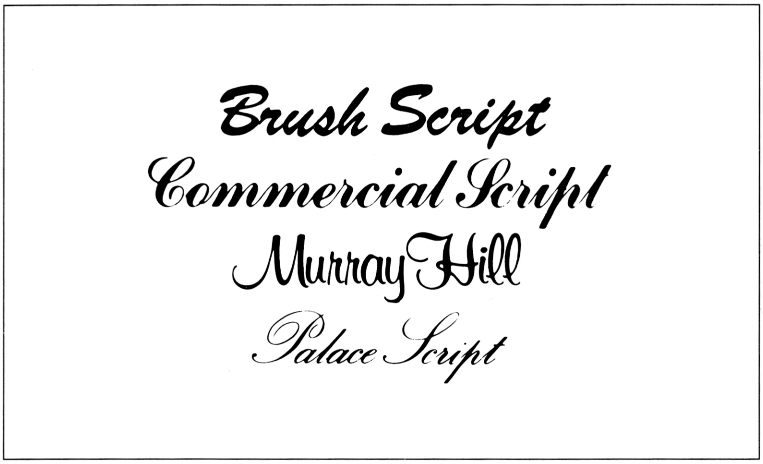

Brush Script
Commercial Script
Murray Hill
Palace Script

Fig. 164
Examples of Scripts

Graphic

These typefaces have been developed from the hand-written forms of the Roman typefaces.

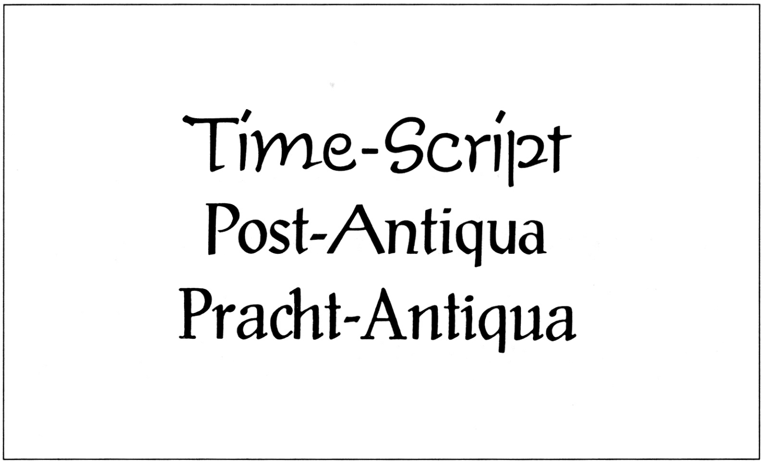

Time-Script
Post-Antiqua
Pracht-Antiqua

Fig. 165
Examples of Graphic

Black Letter

These typefaces mostly have a condensed and irregular lower case. The upper case is decorated with hairlines and has forms full of verve. This group usually does not exist in English classifications.

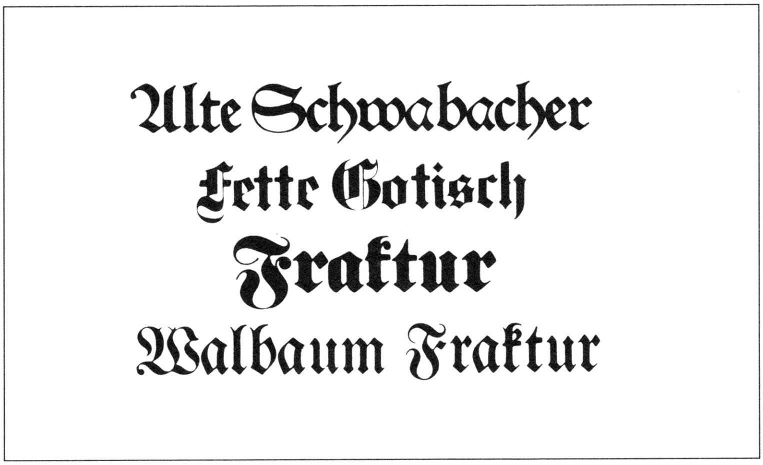

Alte Schwabacher
Fette Gotisch
Fraktur
Walbaum Fraktur

Fig. 166
Examples of
Black letter

Eccentrics

This group does not actually exist, but the author wishes to show typefaces almost illegible.

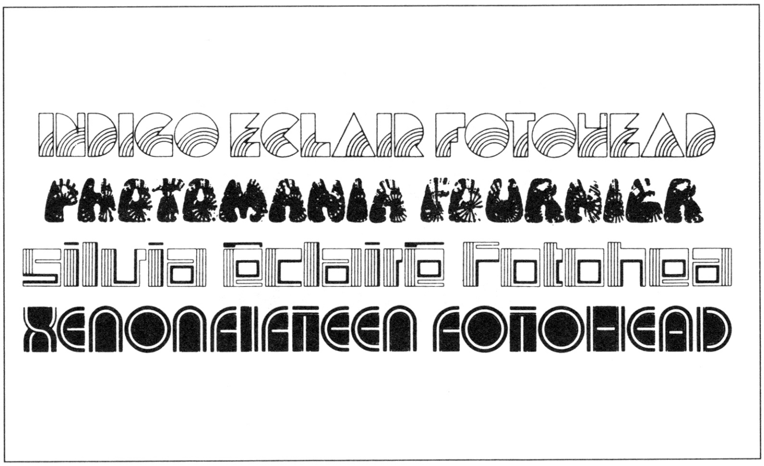

INDIGO ECLAIR FOTOHEAD
PHOTOMANIA FOURNIER
Silvia Eclairé Fotohead
XENONFIFTEEN FOTOHEAD

Fig. 167
Examples of
Eccentrics

Characters of a typeface

This table gives only the most important characters. Refer to the AFII register (Lit. AFII) for detailed descriptions of «glyph» characters.

Upper case

101	A	113	M	125	Y
102	B	114	N	126	Z
103	C	115	O	127	Æ
104	D	116	P	128	Œ
105	E	117	Q	129	Ø
106	F	118	R	201	Ä
107	G	119	S	208	Å
108	H	120	T	210	Ç
109	I	121	U	215	Đ
110	J	122	V	261	Þ
111	K	123	W	237	Ö
112	L	124	X	251	Ü

Fig. 168
URW numbers for upper case

Lower case

301	a	314	n	327	æ
302	b	315	o	328	œ
303	c	316	p	329	ø
304	d	317	q	330	ß
305	e	318	r	331	ı
306	f	319	s	332	ȷ
307	g	320	t	401	ä
308	h	321	u	408	å
309	i	322	v	413	ç
310	j	323	w	436	ö
311	k	324	x	449	ü
312	l	325	y	459	ð
313	m	326	z	460	þ

Fig. 169
URW numbers
for lower case

Numerals

501	1
502	2
503	3
504	4
505	5
506	6
507	7
508	8
509	9
510	0
543	1
544	2
545	3
546	4
547	5
548	6
549	7
550	8
551	9
552	0
661	1⁄1
662	½
663	⅓
664	⅔
665	¼
666	¾
667	⅕
668	⅖
669	⅗
670	⅘
671	⅙
672	⅚
673	⅛
674	⅜
675	⅝
676	⅞
678	0⁄0
679	0⁄00

Fig. 170
URW numbers
for numerals

Punctuation marks, special characters

601	.
602	:
603	·
604	·
605	.
606	…
607	,
608	;
609	’
610	‘
611	”
612	“
613	„
614	!
615	¡
616	?
617	¿
618	»
619	«
620	›
621	‹
622	/
623	-
624	–
625	—
626	(
627	)
628	[
629	]
630	&
631	§
632	†
633	‡
634	*
646	A/S
698	%
699	‰

Fig. 171
URW numbers for punctuation marks

Special characters and floating accents

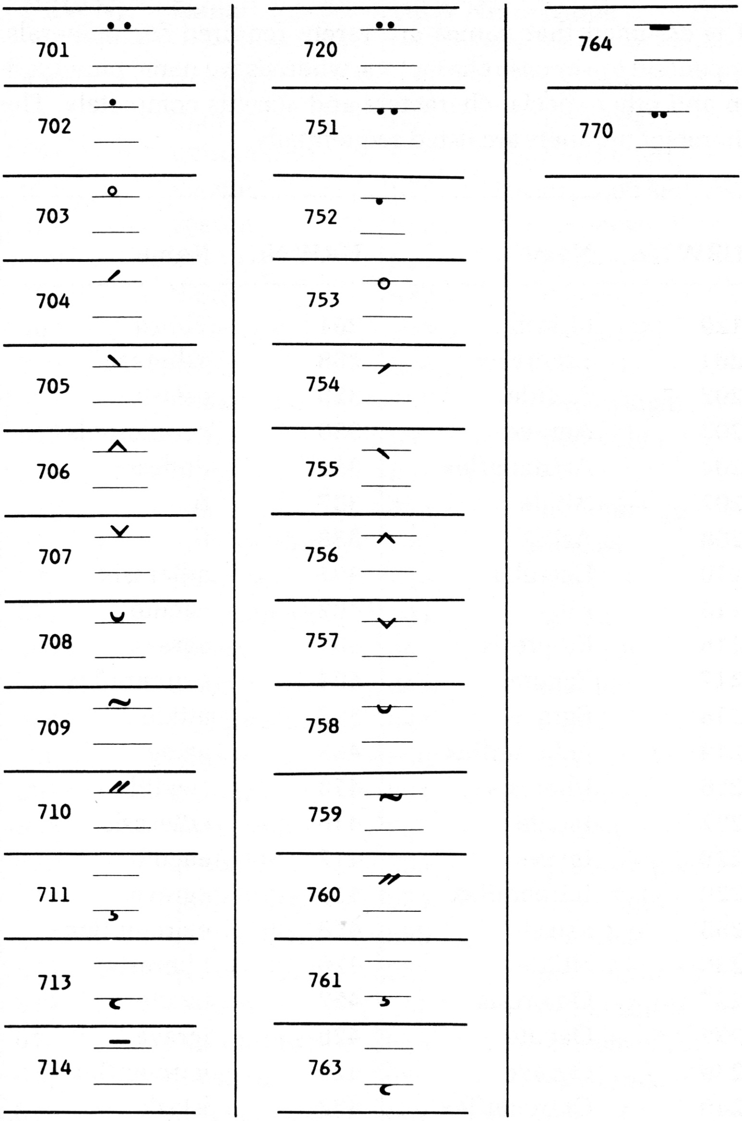

701
702
703
704
705
706
707
708
709
710
711
713
714
720
751
752
753
754
755
756
757
758
759
760
761
763
764
770

Fig. 172
URW numbers for accents

Special character designations

It is assumed that names are rarely required for numerals, upper and lower case characters, whereas we name punctuation and other special characters and accents completely. The character numbers are listed sequentially.

URW Nr.	Name	URW Nr.	Name
129	Oslash	261	Thorn
201	Adieresis	268	Ydieresis
202	Aacute	329	oslash
203	Agrave	330	germandbls
204	Acircumflex	331	dotlessi
207	Atilde	337	fi
208	Aring	338	fl
210	Ccedilla	401	adieresis
215	Eth	402	aacute
216	Edieresis	403	agrave
217	Eacute	404	acircumflex
218	Egrave	407	atilde
219	Ecircumflex	408	aring
226	Idieresis	413	ccedilla
227	Iacute	416	edieresis
228	Igrave	417	eacute
229	Icircumflex	418	egrave
233	Lslash	419	ecircumflex
236	Ntilde	426	idieresis
237	Odieresis	427	iacute
238	Oacute	428	igrave
239	Ograve	429	icircumflex
240	Ocircumflex	432	lslash
241	Otilde	435	ntilde
246	Scaron	436	odieresis
251	Udieresis	437	oacute
252	Uacute	438	ograve
253	Ugrave	439	ocircumflex
254	Ucircumflex	440	otilde
257	Yacute	446	scaron
259	Zcaron	449	udieresis

URW Nr.	Name	URW Nr.	Name
450	uacute	626	parenleft
451	ugrave	627	parenright
452	ucircumflex	628	bracketleft
455	yacute	629	bracketright
457	zcaron	630	ampersand
459	eth	631	section
460	thorn	632	dagger
461	ydieresis	633	daggerdbl
511	sterling	634	asterisk
512	dollar	635	quotesingle
513	cent	636	quotedbl
514	florin	637	at
516	yen	638	numbersign
575	onesuperior	639	degree
576	twosuperior	640	plus
577	threesuperior	641	minus
601	period	642	multiply
602	colon	643	divide
604	periodcentered	644	equal
606	ellipsis	650	trademark
607	comma	651	paragraph
608	semicolon	652	currency
610	quoteleft	653	quotesinglbase
611	quotedblright	654	quoteright
612	quotedblleft	655	braceleft
613	quotedblbase	656	braceright
614	exclam	657	ordfeminine
615	exclamdown	658	ordmasculine
616	question	659	plusminus
617	questiondown	662	onehalf
618	guillemotright	665	onequarter
619	guillemotleft	666	threequarters
620	guilsinglright	677	fraction
621	guilsinglleft	698	percent
622	slash	699	perthousand
623	hyphen	700	backslash
624	endash	751	dieresis
625	emdash	752	dotaccent

URW Nr.	Name	URW Nr.	Name
753	ring	1104	radical
754	acute	1105	integral
755	grave	1109	approxequal
756	circumflex	1111	less
757	caron	1112	greater
758	breve	1113	lessequal
759	tilde	1114	greaterequal
760	hungarumlaut	1117	logicalnot
761	cedilla	1124	infinity
763	ogonek	1133	product
764	macron	1134	summation
795	copyright	1137	partialdiff
796	registered	1151	asciicircum
999	space	1152	bar
999	nbspace	1153	brokenbar
1016	bullet	1154	underscore
1049	lozenge	1199	pi
1059	apple	1337	asciitilde
1085	Delta	1342	mu
1101	notequal	1412	Omega

Appendix C

Accents for Latin type

Albanian	Albanian 3,5	ABCḐEFGHIJKLMNOPQRSTUVWXYZ abcdefghijklmnopqrstuvwxyz ÇçËë «H» “H„
Finnish-Ugrish	Estonian 1	ABCDEFGHIJKLMNOPQRSTUVWXYZ abcdefghijklmnopqrstuvwxyz ÄäÕõÖöÜü (ČčŠšŽž) «H» „H“
	Finnish 4,5	ABCDEFGHIJKLMNOPQRSTUVWXYZ abcdefghijklmnopqrstuvwxyz ÄäÖö ”H” »H» ’H’
	Lappish 0,038	ABCDEFGHIJKLMNOPQRSTUVWXYZ abcdefghijklmnopqrstuvwxyz
	Hungarian 13	ABCDEFGHIJKLMNOPQRSTUVWXYZ abcdefghijklmnopqrstuvwxyz ÁáÉéÍíÓóÖöŐő ÚúÜüŰű „H” »H«
Celtic	Breton 1	ABCDEFGHIJKLMNOPQRSTUVWXYZ abcdefghijklmnopqrstuvwxyz ÉéÈèÊêÏïÑñÜü
	Gaelic 0,08	ABCDEFGHIJKLMNOPQRSTUVWXYZ abcdefghijklmnopqrstuvwxyz ÀàÉéÍíÓóÒòÙù
	Irish 0,4	ABCDEFGHIJKLMNOPQRSTUVWXYZ abcdefghijklmnopqrstuvwxyz ÁáÉéÍíÓóÚú “H” ‘H’ H–H
	Welsh h 0,5	ABCDEFGHIJKLMNOPQRSTUVWXYZ abcdefghijklmnopqrstuvwxyz ÁáÂâÄäÉéÊêËëÎîÏïÔôÖöÙùÛûÜüŴŵẄẅŶŷŸÿ
Germanic	Africaans 4	ABCDEFGHIJKLMNOPQRSTUVWXYZ abcdefghijklmnopqrstuvwxyz „H” ‘H’ H–H ÁáÀàÄäÉéÈèÊêËëÍíÌìÎîÏïÓóÒòÔôÖöÚúÙùÛûÜü
	Danish 5	ABCDEFGHIJKLMNOPQRSTUVWXYZ abcdefghijklmnopqrstuvwxyz ÅåÉéØøÆæ »H« „H“
	German 100–120	ABCDEFGHIJKLMNOPQRSTUVWXYZ abcdefghijklmnopqrstuvwxyz ÄäÖöÜüß „H“ »H« H – H

Taken from Berthold Fototypes EDITION 2.

Fig. 173
Example 1
of accents

	English 350–370	ABCDEFGHIJKLMNOPQRSTUVWXYZ abcdefghijklmnopqrstuvwxyz “H” ‘H’ H–H
	Faroen 0,036	ABCDEFGHIJKLMNOPQRSTUVWXYZ abcdefghijklmnopqrstuvwxyz AáÐðÍíÓóØøÚúÝýÆæ
	Frisian 0,3	ABCDEFGHIJKLMNOPQRSTUVWXYZ abcdefghijklmnopqrstuvwxyz ÂâÉéëïÊêÔôöÚúÛû
	Dutch 19	ABCDEFGHIJKLMNOPQRSTUVWXYZ abcdefghijklmnopqrstuvwxyz „H” "H" ÁáÀàÂâÄäÉéÈèÊêËëÎîÏïÓóÒòÔôÖöÛûÜü
	Icelandic 0,2	ABCDEFGHIJKLMNOPQRSTUVWXYZ abcdefghijklmnopqrstuvwxyz ÁáÐðÉéÍíÓóÖö ÞþÚúÝýÆæ „H“ ‘H’
	Norwegian 4	ABCDEFGHIJKLMNOPQRSTUVWXYZ abcdefghijklmnopqrstuvwxyz ÀàÅåÉéÈèÒòØøÔôÆæ «H»
	Swedish 8,5	ABCDEFGHIJKLMNOPQRSTUVWXYZ abcdefghijklmnopqrstuvwxyz ÅåÄäÉéÖö "H" »H» 'H'
Romanic	French 80–90	ABCDEFGHIJKLMNOPQRSTUVWXYZ abcdefghijklmnopqrstuvwxyz ÀàÂâÇçÉéÈèÊêËëÎîÏïÔôÙùÛûÜüÆæŒœ «H»
	Italian n 60–65	ABCDEFGHIJKLMNOPQRSTUVWXYZ abcdefghijklmnopqrstuvwxyz ÀàÈèÉéÌìÎîÒòÓóÙù «H» “H„
	Catalonian 5	ABCDEFGHIJKLMNOPQRSTUVWXYZ abcdefghijklmnopqrstuvwxyz ÀàÇçÉéÈèÍíÏïÓóÒòÚúÜü
	Latin †	ABCDEFGHIJKLMNOPQRSTUVWXYZ abcdefghijklmnopqrstuvwxyz „H“
	Portugese 125–135	ABCDEFGHIJKLMNOPQRSTUVWXYZ abcdefghijklmnopqrstuvwxyz «H» “H” ‘H’ ÁáÀàÂâÃãÇçÉéÈèÊêÎîÏïÓóÒòÔôÕõÚúÙù

Taken from Berthold Fototypes EDITION 2.

Fig. 174
Example 2
of accents

Romanic	Ratoromanian 0,6	ABCDEFGHIJKLMNOPQRSTUVWXYZ abcdefghijklmnopqrstuvwxyz ÁáÂâÉéÈèÊêÎîÏïÒòÔôÖöÙùÜü
	Rumanian 22	ABCDEFGHIJKLMNOPQRSTUVWXYZ abcdefghijklmnopqrstuvwxyz ĂăÂâÎîŞşŢţ „H“ «H» ‚H‘
	Spanish 220–225	ABCDEFGHIJKLMNOPQRSTUVWXYZ abcdefghijklmnopqrstuvwxyz ÁáÉéÍíÑñÓóÚúÜü ¿H? ¡H! «H» “H” ‘H’ H–H
Baltic	Lithuanian 2,5	ABCDEFGHIJKLMNOPQRSTUVWXYZ abcdefghijklmnopqrstuvwxyz ĄąČčĖėĘęĮįŠšŲųŪūŽž „H“ «H»
	Lettish 1,5	ABCDEFGHIJKLMNOPQRSTUVWXYZ abcdefghijklmnopqrstuvwxyz ĀāČčĒēĢģĪīĶķĻļŅņŖŗŠšŪūŽž «H» „H“
Slawian	Croatic 5	ABCDEFGHIJKLMNOPQRSTUVWXYZ abcdefghijklmnopqrstuvwxyz ĆćČčĐđŠšŽž »H«
	Lower sorbic 0,025	ABCDEFGHIJKLMNOPQRSTUVWXYZ abcdefghijklmnopqrstuvwxyz ČčĚěŁłŃńŔŕŚśŠšŹźŽž „H“
	Upper sorbic 0,025	ABCDEFGHIJKLMNOPQRSTUVWXYZ abcdefghijklmnopqrstuvwxyz ĆćČčĚěŁłŃńÓóŘřŠšŹźŽž „H“
	Polish 34	ABCDEFGHIJKLMNOPQRSTUVWXYZ abcdefghijklmnopqrstuvwxyz ĄąĆćĘęŁłŃńÓóŚśŻżŹź „H“ »H« ‚H‘
	Slovak 4,5	ABCDEFGHIJKLMNOPQRSTUVWXYZ abcdefghijklmnopqrstuvwxyz ÁáÄäČčĎďÉéÍíĹĺĽľŇňÓóÔôŔŕŠšŤťÚúÝýŽž „H“ »H«
	Slowenian 2	ABCDEFGHIJKLMNOPQRSTUVWXYZ abcdefghijklmnopqrstuvwxyz ČčŠšŽž »H« ‚H‘
	Czech 9,5	ABCDEFGHIJKLMNOPQRSTUVWXYZ abcdefghijklmnopqrstuvwxyz ÁáČčĎďÉéĚěÍíŇňÓóŘřŠšŤťÚúŮůÝýŽž „H“ »H«

Taken from Berthold Fototypes EDITION 2.

Fig. 175
Example 3
of accents

Others	Basque 0,7	ABCDEFGHIJKLMNOPQRSTUVWXYZ abcdefghijklmnopqrstuvwxyz ÇçD̄d̄L̄l̄N̄n̄ÑñŔŕR̄r̄T̄t̄Üü
	Esperanto (3–5)	ABCDEFGHIJKLMNOPQRSTUVWXYZ abcdefghijklmnopqrstuvwxyz ĈĉĜĝĤĥĴĵŜŝŬŭ
	Haussa 15	ABCDEFGHIJKLMNOPQRSTUVWXYZ abcdefghijklmnopqrstuvwxyz
	Indonesian 80	ABCDEFGHIJKLMNOPQRSTUVWXYZ abcdefghijklmnopqrstuvwxyz Éé 'H' ‘H’
	Malagassi 4	ABCDEFGHIJKLMNOPQRSTUVWXYZ abcdefghijklmnopqrstuvwxyz ‘H’
	Maltese 0,3	ABCDEFGHIJKLMNOPQRSTUVWXYZ abcdefghijklmnopqrstuvwxyz ÀĊċĠġĦħŻż
	Samoan 0,2	ABCDEFGHIJKLMNOPQRSTUVWXYZ abcdefghijklmnopqrstuvwxyz Ōō
	Suaheli 15	ABCDEFGHIJKLMNOPQRSTUVWXYZ abcdefghijklmnopqrstuvwxyz
	Tagalog 8	ABCDEFGHIJKLMNOPQRSTUVWXYZ abcdefghijklmnopqrstuvwxyz ÁáÀàÂâÉéÈèÊêG̃g̃ÍíÌìÎîÑñÓóÒòÔôÚúÙùÛû
	Turkish 34	ABCDEFGHIJKLMNOPQRSTUVWXYZ abcdefghijklmnopqrstuvwxyz ÂâÇçĞğİıÎîÖöŞşÛûÜü «H» “H„ ‘H’
	Vietnamese 50	ÁÀẢÃẠĂẮẰẲẴẶQ̂Q́Q̀Q̉Q̃ẬĐÉÈẺẼẸÊẾỀỂỄỆÍÌ ỈĨỊÓÒỎÕỌÔỐỒỔỖỘỚỜỞỠỢÚÙỦŨỤỨỪỬỮỰỴ
		áàảãạăắằẳẵặq̂q́q̀q̉q̃ậđéèẻẽẹêếềểễệíìỉĩị óòỏõọôốồổỗộơớờởỡợúùủũụưứừửữựỵ

Taken from Berthold Fototypes EDITION 2.

Fig. 176
Example 4
of accents

Appendix D

Dingbats

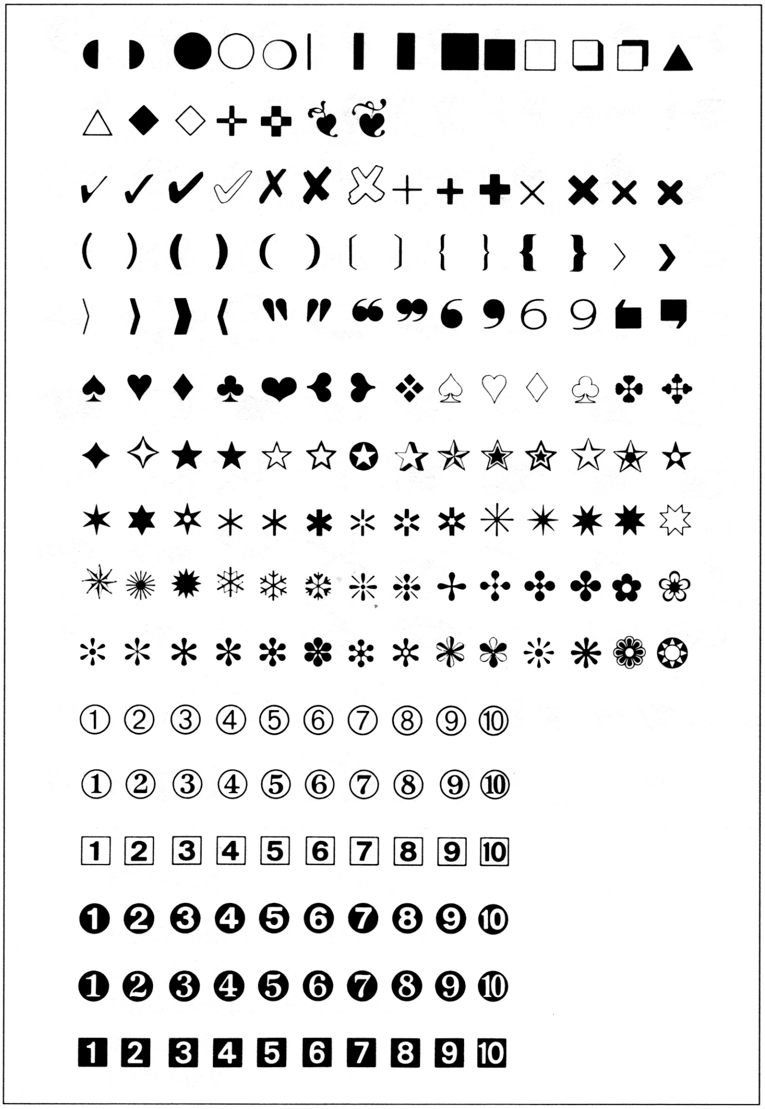


Fig. 177
Example 1
of Dingbats

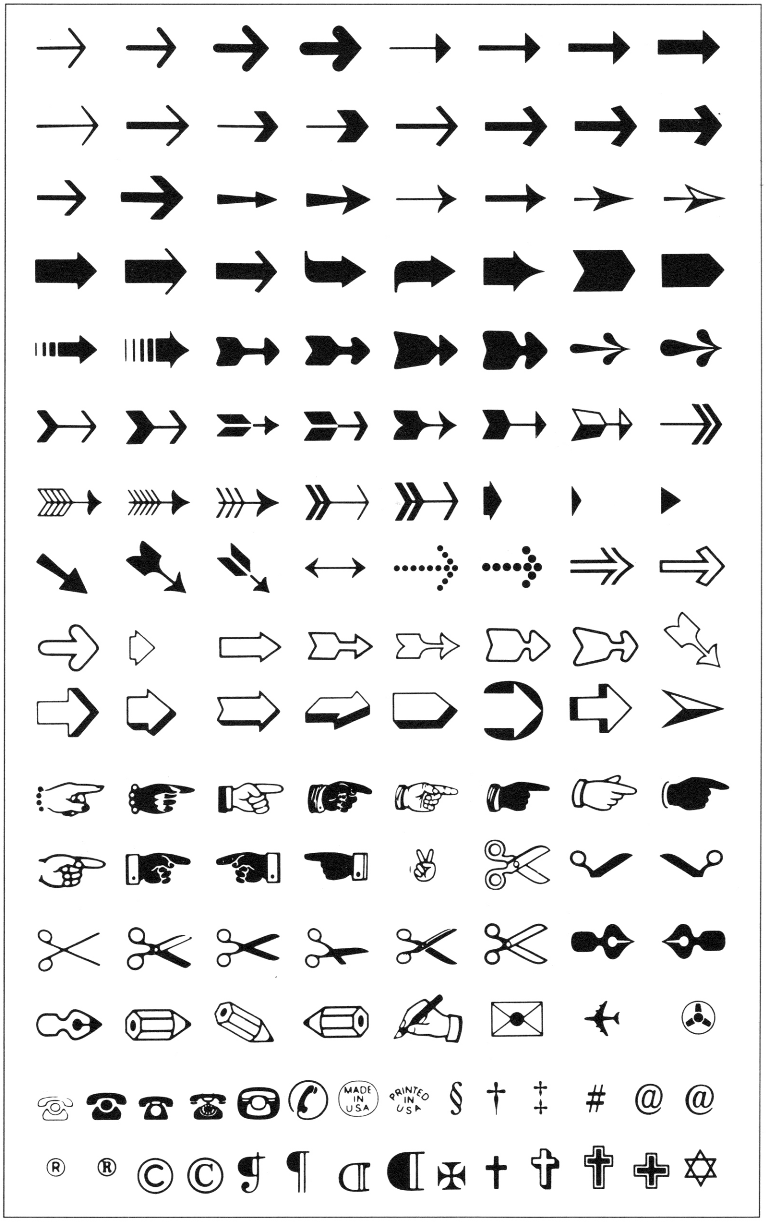

MADE IN USA
PRINTED IN USA

Fig. 178
Example 2
of Dingbats

Explanation of optical effects

We approached approximately 130 people with questions 1 to 5 at URW in Hamburg to obtain the following statistical answers:

Result 1 (squares):	Field b9 ± 2 fields
Result 2 (circles):	Field b9 ± 2 fields
Result 3 (overhangs):	Field b3 ± 1 field
Result 4 (diagonals):	Field a4 ± 2 fields
Result 5 (apices):	Field b2 ± 2 fields

We conclude the following:

- A square appears optically correct when it is 1% ± 1% taller than it is wide.
- A circle appears to be optically correct when it is 1% ± 1% taller than it is wide.
- A circle appears optically as tall as a rectangle when it has an cap and base overhang of 1.5% ± 0.5% respectively.
- A triangle appears optically as wide as a rectangle when it is 5% ± 1% wider.
- A sharp apex (A,V,W) appears to be as high as a straight only when it is 3% ± 1% longer than the square side.

It is possible, of course, that the people tested at URW had a certain unrecognized bias. For this reason, we suggest our readers try the test (page 25) and send the results on a postcard as follows:

“optical appearance”
a7, b9, c4, b3, b2 (example)

to:
URW Software & Type GmbH
attn. Dr. Peter Karow
Harksheider Straße 102
22399 Hamburg
Germany

We guarantee that data and results sent by readers will be used only for the improvement of the statistical base and no other purpose.

Short dictionary of type terminology

a

accent	Akzent (m)	accent (m)
accented (-bearing) letter	Akzentbuchstabe (m)	lettre (f) accentuée
adjustment, «grid fitting»	Einpassung (f), Anpassung (f)	ajustement (m)
administration data	Verwaltungsdaten (pl)	données (fpl) de gestion
aesthetic box	Ästhetiktabelle (f)	table (f) esthétique
alphabet	Alphabet (n)	alphabet (m)
alphabet length	Laufweite (f)	approche (f) de chasse
angle of rotation	Drehwinkel (m)	angle (m) de rotation
angle of italicization	Kursivwinkel (m)	angel d'italisation, inclinaison
to antique	antiquieren	antiquer
arc	Kreisbogen (m)	arc (m) de cercle
arch	Bogen (m)	arc (m)
artwork	Schriftoriginal (n)	original
ascender length, ascender	Oberlänge (f)	ascendante (f)
auxiliary line	Hilfslinie (f)	ligne (f) auxiliaire

b

bar	Querbalken (m), Schwengel (m)	barre (m)
bar serif	Querserife (f)	crochet (f)
base overhang	unterer Überhang (m)	débordement (m) inférieur
baseline	Grundlinie (f), Schriftlinie (f)	ligne (f) de base, ligne (f) d'écriture
bearing value	Fleisch (n)	approche (f)
Bézier function	Bézier-Funktion (f)	fonction (f) de Bézier
bit	Bit (n)	bit (m)
black	schwarz, extrafett	extra gras
black-and-white copy	Schwarzweißvorlage (f)	modèle (m) noir et blanc
black-letter, Gothic	gotische Schrift (f), Fraktur (f)	gotiques (m), caractères (mpl)

to blend together	verschmelzen	fusionner
blending	Verschmelzung (f)	fusion (f)
body matter, text matter, text setting	Textsatz (m)	composition (f) de textes (courants)
body size, type size	Kegel (m), Kegelhöhe (f), Schriftgröße (f)	corps (m), corps de lettre
body type, book type	Brotschrift (f), Werkschrift (f)	caractère (m) courant, caractère (m) de texte
bold	fett	gras
boldness	Strichstärke (f)	grasse (f)
book italic	kursiv Buch	texte italique
bookwork	Werksatz (m), Buchsatz	composition (f) de travaux de labeur
border	Rahmen (m)	cadre (f), bordure (f)
bounding box	umhüllendes –, umschreibendes Rechteck (n)	boîte (f) englobante
bow	Rundung (f), Bogen (m)	arrondi (m), panse (f)
broken	gebrochen	fracturé
byte	Byte (n)	octet (m)

c

CAD, computer aided design	CAD (n)	Dessin (m) Assisté par Ordinateur, DAO
calculation	Berechnung (f)	calcul (m)
calligraphy	Kalligraphie (f)	calligraphie (f)
cap height	Versalhöhe (f), Majuskelhöhe (f)	hauteur (f) des capitales, hauteur des majuscules
capital letter, cap, upper case letter, large letter	Majuskel (f), Versalie (f) Kapitalbuchstabe (m), Großbuchstabe	majuscule (f), capitale (f), haut (m) de casse, grande lettre (f)
cassette	Kassette (f)	cassette (f)
cathod ray tube, CRT	Kathodenstrahlröhre (f) (CRT), Fernsehröhre (f)	tube (m) à rayons cathodiques
channel processing, rectangular alignment	Kanalautomatik (f)	canalisation (f), alignement (m) automatique
character, letter	Schriftzeichen (n), Buchstabe (m)	caractère (m) signe (m) typographique
character style	Buchstabenform (f)	form (f) de lettre, style (m)
chord	Sehne (f)	corde (f)
circle	Kreis (m)	cercle (m)
circle segment	Teilkreis (m)	segment (m) de cercle

circle vector, C vector	Kreisvektor (m), C-Vektor (m)	vecteur (m) courbe, vecteur Cb
circular	kreisförmig	circulaire
closed contour	geschlossene Kontur (f)	contour (m) fermé
coarse resolution, low resolution	grobe Auflösung (f)	basse résolution (f), résolution (f) grossière
coding technique	Kodierungstechnik (f)	technique (f) de codage
column	Spalte (f)	colonne (f)
command code	Befehlskode (m)	code (m) de commande
composition	Satz (m)	composition (f)
computer	Computer (m)	ordinateur (m)
concave foot, hollow-cast foot	Hohlfuß (m)	pied (m) creux, pied cavé
to condense	kondensieren	comprimer
condensed	schmal	étroite
to contour	konturieren	dessiner le contour
contour	Kontur (f)	contour (m)
contouring	Konturieren (n)	silhouettage (m)
control signal	Steuersignal (n)	signal (m) de commande, signal de contrôle
to convert	wandeln	convertir
coordinate	Koordinate (f)	coordonnée (f)
coordinate system	Koordinatensystem (n)	système (m) de coordonnées
coordinate zero point, origin	Koordinatennullpunkt (m), Ursprung (m)	point (m) zéro de coordonnées, point d'origine
copper plate	Kupferdruckplatte (f)	plaque (f) de cuivre, planche (f) d'impression pour taille-douce
corner point	Eckpunkt (m)	point (m) d'angle
counter, white inner space	Punze (f), weißer Innenraum (m)	contre-poinçon (m) boucle (f)
covering	Überdeckung (f)	recouvrement (m)
cross stem, crossbar	Querbalken (m)	barre (f) transversale
cross-hair	Fadenkreuz (n)	réticule (m)
Cursive	Handschrift (f)	script , écriture manuscrite
cursor	Kursor (m), Sensor (m)	curseur (m)
curve	Kurve (f), Kurvenlinie (f)	courbe (f)
curve adjustment	Kurvenausgleich	compensation des courbes
curve bar	Brücke (f)	pont (m)
curve point	Kurvenpunkt (m)	point (m) de courbe
curve stem	Kurvenbalken (m)	fût concave (m)

curve stroke	runder Balken (m)	trait (m) courbe, barre arrondie
curved line	runder Strich (m)	trait (m) courbe
cutting foils	Schneiden von Folien (n)	découpe (f) de feuilles
cutting machine	Schneidemaschine (f)	machine (f) à couper

d

dancing	tanzend	valsante
data acquisition	Datenerfassung (f)	saisie (f) de données
data administration	Datenverwaltung (f)	gestion (f) de données
data bank	Datenbank (f)	banque (f) de données
data base	Datenbasis (f)	base (f) de données
data buffers	Datenpuffer (m)	mémoire (f) tampon
data carrier, data carrying media	Datenträger (m)	support (m) de données
data processing	Datenverarbeitung (f)	traitement (m) de l'information
decorative typeface	Zierschrift (f)	alphabet (f) gracile
deformation	Verformung (f)	déformation (f)
delta	Delta (n)	delta (f)
demi, demi bold	halbfett	demi-gras
descender length, descender	Unterlänge (f)	descendante (f)
diagonal	Diagonale (f)	diagonale (f)
Didot point, p	Didot-Punkt (m)	point (m) didot
digital	digital	digital, numérisé
digital coding technique	digitale Kodierungsart (f)	type (m) de codage digital
digital representation	digitale Darstellung (f)	représentation digitale (f)
digitization	Digitalisierung (f)	digitalisation (f), numérisation (f)
digitizer	Digitizer (m)	digitaliseur (m), tablette (f) à numériser
direct VDT	Speichersichtgerät (n)	écran (m) vidéo direct
disk cartridge	Wechselplattenkassette (f)	cartouche (f)
diskette, floppy disk	Diskette (f), Floppy (f)	disquette (f), floppy (m)
displacement	Verschiebung (f)	déplacement (m)
display device, visual display terminal (VDT)	Sichtgerät (n)	écran (m) de visualisation
display type	Auszeichnungsschrift (f)	caractère (m) de titre, caractère fantaisie, caractère distinctif
dot, pixel	Bildpunkt (m), Pixel (n)	point (m) d'image, pixel

dot-matrix-printer	Schnelldrucker (m)	imprimante (f) rapide
dots per inch (dpi)	Punkte pro inch (dpi), (plur., m)	points par pouce (dpi) (mpl)
down-stroke	Abstrich (m)	plein (m)
drawing	Zeichnung (f)	dessin (m)
drawing machine	Zeichenmaschine (f)	machine (f) à dessiner
dropout	Mindeststärke (f)	épaisseur minimale (f)
DTP, Desk Top Publishing	DTP, Desk Top Publishing (n)	Publication (f) Assisté par Ordinateur, PAO
ductus	Duktus (m), Strichführung	ductus (m)

e

ear	Ohr (n)	oreille (f)
edge	Rand (m)	bordure (f), contour (m)
electroerosion printer	Elektro-Erosionsdrucker	imprimante (f) à électro-érosion
electron beam, light beam	Elektronenstrahl (m), Lichtstrahl (m)	faisceau (m) d'électrons, faisceau électronique
electronic storage (media)	elektronischer Speicher	support (m) électronique
electrostatic printer	elektrostatischer Drucker	imprimante (f) électrostatique, traceur (m) électrostatique
em resolution, resolution for the em square	Geviertauflösung (f), Gitter(netz) (n)	résolution (f) de cadratin, quadrillage (m)
em-square, em	Geviert (n)	cadratin (m)
end point	Endpunkt (m)	point (m) d'arrivée
engraving machine	Graviermaschien (f)	machine (f) à graver
to enlarge	vergrößern	agrandir
enlargement	Vergrößerung (f)	agrandissement (m)
to expand	expandieren	élargir
expanded, large	breit	élargi, large
exponent	Exponent (m)	exposant (m)
to expose	belichten	impressionner
exposing films	Filmbelichtung (f)	exposition (f) de films
extension	Auszug	sortie (f) gracile, paraphe (m)
extra bold, black	extra fett	extra gras
extra light	extra leicht	extra maigre
extreme value of the curve	Kurvenextremum (n)	extrémité (f) de courbe

f

file	Datei (f)	fichier (m)
fixed accent	Fixakzent (m), Festakzent	accent (m) fixe

flat bed drawing machine	Flachbettzeichenmaschine	table (f) traçante
floating accent	fliegender Akzent (m)	accent (m) flottant
foil cutting machine	Folienschneidemaschine	machine (f) à découper des feuilles
to focus	fokussieren	focaliser
foot-note, bottom note	Fußnote (f)	note (f) en bas de page, note courante, renvoi (m)
form setting	Formsatz (m)	composition (f) formée
formula setting	Formelsatz (m)	composition (f) de formules
foundry	Schriftguß (m), Gießerei (f)	fonderie (f)
fraction	Bruch (m)	fraction (f)
frame	Rahmen (f)	ossature (f)

g

graphic	Graphik (f)	forme (f) graphique, graphisme (m)
graphic display device, graphic VDT	graphisches Sichtgerät (n), Graphic VDT (n)	terminal (m) vidéo graphique, écran (m) graphique
grid	Gitter (n)	grille (f)
grotesque	grotesk	style (f) grotesque, (sans patin)

h

hairline	Haarlinie (f)	délié (m)
hand-digitization	Handdigitalisierung (f)	digitalisation (f) manuelle
harddisk	Festplatte (f)	disque (m) dur
hardware	Hardware (f)	matériel (m)
headline	Headline, Überschrift (f)	titre
high resolution	feine (hohe) Auflösung (f)	haute résolution (f)
host computer	Host-Rechner (m)	unité (f) centrale, ordinateur (m) principal
hot metal composition	Bleisatz (m)	composition (f) plomb

i

illuminated advertising	Licht-, Leuchtreklame (f)	enseigne (f) lumineuse
inclination	Schrägung (f), Schräge (f), Neigung (f)	inclinaison (f), oblique (f)
indentation	Einzug (m) im Text	renfoncement (m)
index	Index	index (m)
index sequentiell access mode	ISAM-Zugriff (m)	mode (m) d'accès séquentiel indexé

initial	Initial	lettre (f) initiale
ink jet printer	Inkjet-Drucker (m)	imprimante (f) à jet d'encre
ink trap	Einzug (m)	renfoncement (m)
inner contour, inner edge	Innenkontur (f), Innenrand (m)	contour (m) intérieur contour (m) interne
input	Texteingabe (f), Eingabe (f)	saisie (f) des textes, entrée (f)
interpolation	Interpolation (f)	interpolation (f)
intersection	Schnittstelle (f)	intersection (f)
italic	kursiv	italique
italic angle, slanting angle	Kursivwinkel (m)	angle (m) d'inclinaison, angle italique
to italicize	kursivieren	pencher, italiser
italicizing	Kursivierung (f)	italisation (f)

j

job composition, jobbing composition	Akzidenzsatz (m)	composition (f) de travaux de ville,
job printing	Akzidenzdruck (m)	travaux (plf) de ville
job type, jobbing typeface	Akzidenzschrift (f)	caractère (m) (pour) travaux de ville, caractère fantaisie
join	Einlauf	entrée (f)

k

kerning	Unterschneiden (n), Kerning (n)	crénage (m)
kerning table	Unterschneidungstabelle, Kerningtabelle (f)	table (f) de crénage
keyboard	Tastatur (f)	clavier (m)
knot	Knoten (m)	nœud (m)

l

laser	Laser (m)	laser (m)
laser beam	Laserstrahl (m)	faisceau (m) laser rayon (m) laser
laser printer	Laserdrucker (m)	imprimante (f) à laser
laser vaporization machine	Dry-Transfer-Maschine (f), Laserverdampfer (m)	machine (f) à vaporisation laser (m)
layout	Entwurf (m), Layout (n)	maquette (f)
leading	Durchschuß (m)	interlignage (f)
left side bearing	Vorbreite (f)	approche (f) de gauche

legibility	Lesbarkeit (f)	lisibilité (f)
letter, character	Buchstabe (m)	lettre (f) , caractère (m)
(letter) index table	(Buchstaben)-Adreßbuch	répertoire (m) (de lettres)
letterpress printing	Buchdruck (m)	impression (f) typogaphique
letter space	Buchstabenzwischenraum	espace (m)des caractères
letter spacing	Spationierung (f)	interlettrage (f)
ligature	Ligatur (f)	ligature (f), lettres (fpl) accouplées
light	mager	maigre
line	Linie, Zeile (f)	ligne (f)
line element	Linienelement (n)	filet (m)
line graphique	Strichgraphik (f)	graphique (m) au trait
line resolution	Linienauflösung (f)	résolution (f) par ligne
line spacing, leading	Zeilenabstand (m), Durchschuß (m)	interlignage (f), écartement (m) des lignes
link	Steg, (m) Verbindung (f)	liaison (f)
loading	Laden (n)	chargement (m)
logo	Logo (n)	logo (m)
long vector, L vector	langer (L-)Vektor (m)	vecteur (m) long, vecteur L
loop	Schleife (f)	boucle (f)
lower	tiefsetzen	abaisser
low resolution	niedrige Auflösung (f)	basse résolution (f)
lower body line	Kegelunterkante (f)	limite (f) inférieure du corps
lower case letter, minuscule, small letter	Minuskel (f), Kleinbuchstaben (m), Gemeine (m)	minuscule (f), bas de casse (f), petite lettre (f)
lower index, inferior, subscript	Index (m)	indice (m) inférieur

m

machine format	Maschinenformat (n)	format (m) de machine
magnetic tape	Magnetband (n)	bande (f) magnétique
manual correction	Handkorrektur (f)	correction (f) manuelle
marking	Markierung (f)	marquage (m)
matrix	Matrix (f)	matrice (f)
matrix printer	Matrixdrucker (m), Nadeldrucker (m)	imprimante (f) matricielle imprimante (f) à aiguilles
mean line	Minuskelhöhe (f)	hauteur (f) des minuscule
medium	normal	normal
memory capacity requirement	Speicherbedarf (m)	besoin (m) de mémoire

menu	Menü (n)	menu (m)
menu field	Menüfeld (n)	boîte-menu (m)
micro filming device	Mikroverfilmungsgerät (n)	appareil (m) à microfilmer
microcomputer	Microcomputer (m)	micro-ordinateur (m)
microfilm	Mikrofilm (m)	microfilm (m)
mirroring	spiegelnd	effet (m) miroir
modification	Modifikation (f)	modification
monospaced	dicktengleich	chasse (f) de fixe

n

NC machine, numerically controlled machine	NC-Maschine (f)	machine (f) à commande numérique
numeral	Ziffer (f)	chiffre (m)
numerically controlled	numerische Steuerung (f)	commande (f) numérique, commande digitale

o

offset, (lithographic) offset printing	Offsetdruck (m)	impression (f) offset
open contour	offene Kontur (f)	contour (m) ouvert
operating system	Betriebssystem (n)	système (m) d'exploitation
optical character recognition, OCR	optische Zeichenerkennung (f)	reconnaissance (f) optique des caractères
original	Schriftoriginal (n)	origine (f)
ornamental letter, swash-character, eccentric character, fancy letter	Zierbuchstabe (m), exzentrisches Schriftzeichen (n)	caractère (m) ornemental, caractère gracile, caractère excentrique
ornamentation	Verzierung (f)	ornement (m)
outer contour, outer edge	Außenkontur (f), Außenrand (m)	contour (m) extérieur, contour externe
outline, contour	Kontur (f)	contour (m)
output	Druck (m), Ausgabe (f)	impression (f), sortie (f)
overhang	Überhang (m)	déborde- (m), dépassement
overlapping table	Überdeckungstabelle (f)	table (f) de chevauchement

p

page make-up	Seitenumbruch (m), Seitenlayout (n)	mise (f) en page
pattern	Vorlage (f)	modèle (m)
perception	Anmutung (f)	

performance specification	Pflichtenheft (n)	cahier (m) des charges
personal computer, PC	PC, Personal Computer	ordinateur (m) personnel
photocomposer, photocomposing machine	Lichtsetzgerät (n), Lichtsatzmaschine (f)	photocomposeuse (f)
phototypesetting	Photosatz (m)	photocomposition (f)
pica point, pt	Pica-Punkt (m)	point (m) pica
pictorial sign	Piktogramm (n), Bildzeichen (n)	signe (m) pictographique, pictogramme (m)
picture transmission	Bildübertragung (f)	transmission (f) d'images
pixel	Rasterpunkt (m), Pixel (m)	pixel (m), point (m) de trame
plan	Plan (m)	plan (m)
plotter	Plotter (m)	traceur (m)
point of inflection	Wendepunkt (m)	point (m) d'inflexion
pointer	Zeiger (m)	pointeur (m)
pointsize, dot size	Schriftgrad (m), Schriftgröße (f), Punktgröße (f)	dimension (f) en points corps (m)
post processor	Postprozessor (m)	post-processeur (m)
poster	Poster (n), Plakat (n)	poster (m), placard (n)
poster type	Poster- (d), Plakatschrift (f)	caractère (m) de poster
printing beam	Schreibstrahl (m)	rayon (m) d'impression
printing plate	Druckplatte (f)	plaque (f) d'impression
programming	Programmierung (f)	programmation (f)
proof printing	Proof-Printing (n), Andruck (m)	impression (f) des épreuves
proof, proof print	Fahne (f), Korrekturabzug	placard (m), épreuve (f)
punch-cutter	Stempelschneider (m)	graveur (m) de poinçons
punctuation mark	Satzzeichen (n), Interpunktion (f)	signe (m) de ponctuation

q

quadrant	Quadrant (m)	cadran (m)

r

to raise, to elevate	hochsetzen	surélever
RAM,	RAM-speicher (m)	mémoire (f) vive (RAM)
raster	Raster (n)	trame (f)
raster point	Rasterpunkt (m)	point (m) de frame
raster resolution	Rasterauflösung (f)	résolution (f) de trame
raster scan VDT (visual display terminal)	Rastersichtgerät (n)	écran (m) vidéo tramé
rastering	Aufrasterung (f)	tramage (m)

readability	Leserlichkeit (f)	lisibilité (f), déchiffrage (m)
record	(Daten)-Satz (m)	record (m), enregistrement (m)
rectangular alignment	Kanalverarbeitung	alignement (m) automatique
to reduce	verkleinern	réduire
reduction	Verkleinerung (f)	réduction (f)
resolution	Auflösung (f)	résolution (f)
resolution mark	Spitze (f)	pointe (f)
right side bearing	Nachbreite (f)	approche (f) de droite
Roman monumental lettering	Römische Lapidarschrift (f)	lapidaire (f) romaine
Roman (typeface)	Antiqua (Schrift) (f)	romain (m), elzévir
to rotate	rotieren	faire une rotation
rotation	Drehung (f), Rotation (f)	rotation (f)
to round	runden	arrondir
routing machine, drilling machine	Fräsmaschine (f)	machine (f) à fraiser, fraiseuse (f)
row	Zeile (f)	rangée (f), ligne (f)
run length	Lauflänge (f)	ligne (f), tracé (m) de ligne
run length code	Lauflängenkodierung (f)	codage des parcours
running width	Laufweite (f)	approche (f) de chasse

s

sans-serif type, Sans serif	Grotesk(schrift) (f), serifenlose Schrift (f), Linear-Antiqua (f)	antiques (fpl), caractères (mpl) sans empattement(s), écriture (f) grotesque
sansserif	serifenlos	sans empattements
to scan	scannen, abtasten	scanner, balayer
scan data	Scandaten (pl)	données (fpl) scannées
scanline	Scanlinie (f), Bildlinie (f)	ligne (f) scannée, ligne de balayage, ligne (f) d'image
scanner	Scanner (m)	scanner (m)
scanning	Aufrasterung (f)	scanning (m)
script type, Script	Schreibschrift (f)	cursives (fpl), calligraphiques (fpl), écriture (f)
segmentation	Elementzerlegung (f)	segmentation (f)
self adhesive material	Klebefolie (f)	matériel (m) adhésif
semiconductor memory	Halbleiterspeicher (m)	mémoire (f) à semiconducteurs
sensor key	Sensortaste (f)	touche (f) du curseur
serif	Serife (f)	empattement (m), sérif (m), patin (m)

serif type	Antiqua (Schrift) (f)	style (m) romain
set-width line	Dicktenleiste (f)	liteau (m) de chasse
setting	Satz (m)	composition (m)
sexy spacing	sexy spacing	sexy spacing
to shadow	schattieren	ombrer
shaping	verformend	déformée
short vector, S vector	kurzer (S) Vektor (m)	vecteur (m) court, vecteur Ct
shoulder	Schulter	épaule
sign	Schild (n)	panneau (m)
to size	umgrößern	mettre à la cote
slant	Schräge (f)	inclinaison (f)
sloped stem	schräger Balken (m)	barre (f) oblique, ≈inclinée
small cap	Kapitälchen (m)	petite capitale (f)
to smooth	glätten	lisser
software	Software (f)	logiciel (m)
spacing (letter -)	Zurichtung (f)	chasse (f) de caractères
sparkling (bouncing)	tanzend	valsante
special setting	Spezialsatz (m)	chasse (f) spéciale
spiral	Spirale (f)	spirale (f)
spline curve	Spline-Kurve (f)	courbe (f) spline
spot	Fleck (m)	tache (f), point (m)
spur	Sporn (m)	éperon (m)
square	Quadrat (m)	carré (m)
start point	Startpunkt (m)	point (m) de départ
stem	Balken (m)	barre (f), trait (m), fût (m)
stem edge	Balkenrand (m)	bornéè (f)
stem width	Balkenstärke, Strichstärke	épaisseur (m) de trait
step-like image	treppenartiges Bild (n)	structure (f) en escalier
storage	Speicherung (f)	enregistrement (m), mèmorisation (f), stockage (f)
storage requirement	Speicherkapazität (f)	capacité (f) mémoire
to store	speichern	mémoriser, stocker, enregistrer
straight	Gerade (f)	droite (f), ligne (f) droite
straight line	gerader Strich (m)	trait (m) vertical
straight stem	gerader Balken (m)	barre (f) droite, fût (m)
stress	Verdickung (f)	épaissement (m)
stretching	streckend	extensible
stroke	Strich (m)	trait (m)
stroke thickness	Strichstärke (f), Strichdicke (f)	épaisseur (f) de trait
stroke width	Strichbreite (f)	largeur (f) de trait

superior, superscript, exponent	Index obenstehender (m), Exponent (m)	exposant (m)
swash character	Zierbuchstabe (m), Schwungbuchstabe (m)	lettre (f) ornée
swash, embellished stroke	Zierstrich (m)	trait (m) ornemental, trait gracile
symbol	Signet (n)	symbole (m)

t

table	Tabelle (f)	tableau (m)
tabular setting	Tabellensatz (m)	composition (m) de tableau
tail	Abschwung, Schwanz (m)	queue (f)
tangent point	Tangentenpunkt (m)	point (m) de tangente
tangential transition	tangentialer Übergang (m)	passage (m) tangentiel
temporary memory	temporärer Speicher (m)	mémoire (f) temporaire
tension	Spannung (f)	tension (m)
terminal	Endung (f)	sortie (f)
text type, text typeface	Brotschrift, Textschrift	caractères (m) de texte
thickening	Verdickung (f)	épaississement (m), grossissement (m)
tight setting, narrow set	enger Satz (m)	composition (f) serrée
total width	Dickte (f)	chasse (f)
touching	berührend	d'acollage
touching table	Berührungstabelle (f)	table (f) de contact
traditional	traditionell	traditionelle
transformation	Transformation (f)	transformation (f)
transition	Übergang (m)	transition (f)
trapping, shrink & spread	Absparen (n), Überfüllen	recouvrement (m)
triangle	Rechteck (n)	rectangle (m)
type casting	Schriftguß (m)	fonte (f) des caractères
type designer	Schriftgestalter (m), Schriftkünstler (m), Schriftschöpfer (m)	dessinateur (m), créateur (m), concepteur (m) des caractères
type, typeface, font	Schrift (f)	police (f) des caractères, fonte (f), style (m)
type height	Schrifthöhe (f)	hauteur (f) des caractères
typesetting machine	Setzmaschine (f)	composeuse (f), flasheuse (f)
type size	Schriftgrad (m)	corps (m)
type style	Schriftstil (m)	style (m) de caractère
typewriter composition	Schreibsatz (m)	composition (f) par machine à écrire, composition par frappe directe

typography	Typographie (f)	typographie (f)

u

uncial script	Unziale (f), Unzialschrift (f)	onciale (f), caractères onciaux (m)
unit (size)	(Maß)einheit (f)	unité (f) (de mesure)
unit vector	Einheitsvektor (m)	vecteur (m) unitaire
upper overhang	oberer Überhang (m)	débordement (m) supérieur
upright	gerade, geradestehend	droit, romain
up-stroke	Aufstrich (m)	delié (m)

v

vector	Vektor (m)	vecteur (m)
vector format	Vektorformat (n)	format (m) vectoriel
Venetian Serif	Venetianische Antiqua (f)	Vénitienne (f)
version	(Schrift)schnitt (m)	version (f)
video communication system	Bildschirmtext (Btx) (m)	systèmes (m) de communication vidéo
vignette, embellishment	Vignette (f)	vignette (f), enjolivure (f)
visual display device, VDT	Sichtgerät (n), VDT (n)	terminal (m) vidéo (VDT), poste (m) avec écran de visualisation

w

weight (designation)	Fettegrad (m)	graisse (f)
white ink trap	weißer Einzug (m), Einschnitt (m)	encoche (f)
white space	Zwischenraum (m)	espace (m) blanc
white vector, W vector	Weißvektor (m), W-Vektor (m)	vecteur (m) blanc, vecteur B
width	Dickte (f)	chasse (f)
width table	Dicktentabelle (f)	table (f) de chasse
word-spacing	Wortabstand (m)	espacement (m) de mots
workstation	Arbeitsstation (f)	poste (m) de travail
writing device	Schreibgerät (n)	dispositif (m) d'écriture
writing resolution	Schreibdichte (f)	densité (f) d'écriture

x

x-height	Minuskelhöhe (f)	hauteur (f) des minuscules

Data structure for typefaces

Fonts are stored digitally by a vast number of different methods, as we have shown. These often very different sorts of data lead to different formats, each presenting character image information in a different manner. Despite this, it is possible to find a uniform structure for data on a data carrier. Following, the structure used by URW is described, applicable to all fonts supplied irrespective of format. The formats themselves are described individually in the appropriate appendices H to Q.

Data carrying media for font data

Several years ago, URW preferred magnetic tape as medium to deliver typefaces to customers, because it was the only standard supported by nearly all vendors of computers and operating systems.

Today this has changed. Magnetic tapes are out of fashion. Instead, there is a wide variety of data carriers, which can be written on one system and read on another one. Take floppy-disks, written on a Macintosh or PC and read on Unix workstations. On the other hand, nearly all companies have a manifold of computers from different manufactures, usually all integrated in a network. So no matter, what data carrier is sent, it can be written or read on one of the workstations and the data are sent to this final destination in the network.

So we have the situation, that data are stored on a variety of media, using the utilities and the file system of the corresponding operating system.

Structure of a font file

Each font is stored in a file corresponding to the filesystem of the operating system. But the internal structure of an URW font file does not depend on the environment. It is uniform for most of the different font formats and consists basically of a font header and records for all the characters in the font.

The font header is detailed and the character record comprehensively described in the following. Within the terms of the character record the contour index is only applicable to IK, DI, VC, VS and VE formats. The SC, SN, BI and GS formats have no contour indices.

The image information is described generally here, and detailed in the appendices of individual formats. The unit for coordinates is normally 1/100 mm or 1 pixel, in which the identifier for the measurement unit is set to 1 (see below). The quantity «word» refers to a 16 bit variable. «Record» means 2,048 words, a more or less historical relic, when IKARUS only worked on VAX computers.

Character numbers progress from 1 to 65,535. At URW numbers 1 to 999 are reserved for Latin fonts, while Kanji fonts generally require use of more than 1,000 numbers.

Font header

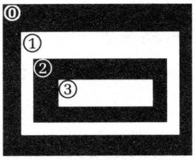
Word	Meaning	Comment
1	Length of font header in words	= 1 + I + J + L + M
	Name section	
2	Length of name section in words	(I)
3	Number on URW list	
4–9	**File name**	6 words
10–49	Font name in ASCII	40 words
50	Format ident of data	e.g. IK, DI, G1...G4, VC, ...
51–53	Production date	3 words
54–56	Date of last change	e.g. 16, 6, 93 for 16th of June 1993 Currently (1993) I has the value 55 at URW.
	Font information	
I + 2	Length of font data in words	(J)
I + 3	Ident of font	= 1 for typeface
I + 4	**Number of characters**	
I + 5	Cap height	
I + 6	Body size	
I + 7	x-height	given in 1/100 mm or in pixels
I + 8	Distance of base line from lower body line	
I + 9	Line distance for composition	
I + 10	Stroke thickness	
I + 11	Italic angle	in 1/10 degrees
I + 12	Optimum size for composition in Didot points	e.g. 12 p

Word	Meaning	Comment
I + 13	Average character width (based on international character frequency)	in 1/100 mm At URW (1993) J currently has the value 12.
	Hierarchy section	
I + J + 2	Length of hierarchy section in words	(L) = 1 for fonts
	Character index	
I + J + L + 2	Length of character index in words	(M)
I + J + L + 3	Pointer to last written record (= length of font in records)	
I + J + L + 4	Pointer to last written word of last record	
I + J + L + 5	Number of first character	3 words for one character
I + J + L + 6	Pointer to record in this font which contains the start of the first character	
I + J + L + 7	Pointer in this record to the word which is the first word of the corresponding character record	
I + J + L + 8	Number of second character etc.	
.	.	
.	.	
.	.	
I + J + L + M + 1	Pointer to the word of the character record where the last character starts	

Character record

Word	Meaning	Comment
	Length of first character	2+N1+N2+N3+M
1	in records	
2	in words	
	Name section	
3	Length of name section in words	N1 (currently=3)
4	Character number	1, ….. , 65535
5	Reserved	0
	Setting information	
N1 + 3	Length of setting information	N2 (currently = 12)
N1 + 4	Character Type	1 = letter 2 = logo/signet 3 = line graphic 4 = frame
N1 + 5	Number of digitizations	
N1 + 6	Total width (**T**)	T = L + W + R given in units
N1 + 7	Left side bearing (**L**)	
N1 + 8	Width (**W**)	
N1 + 9	Right side bearing (**R**)	

Word	Meaning	Comment
N1 + 10 N1 + 11 N1 + 12 N1 + 13	X minimum X maximum Y minimum Y maximum bounding box	given in units
N1 + N2 + 2	Unit size (definition of the unit)	= 1 ≙ unit is 1/100 mm or ≙ 1 pixel **(normal)** = 2 ≙ unit is 2/100 mm = 3 etc. When the unit is set to 2, for example, one must multiply the coordinates prior to the use of the character by a factor of 2.
	Contour index	
	Length of contour index	N3=N4×2048+N5
N1 + N2 + 3	in records (N4)	
N1 + N2 + 4	in words (N5)	
N1 + N2 + 5	Record pointer,	
N1 + N2 + 6	Word pointer of first contour relative to start of image information. The start of the image information relative to character data record is **3 + N1 + N2 + N3.**	
N1 + N2 + 7	Sense of rotation of first contour	– 1 = clockwise 0 = open contour + 1 = anti-clockwise

Word	Meaning	Comment
N1 + N2 + 8	Identifier for nesting hierarchy	0 = outer contour 1st order 1 = inner contour 1st order 2 = outer contour 2nd order 3 = inner contour 2nd order 4 = etc. 
N1 + N2 + 9	Color inside contour	0 = transparent 1 = black, etc.
N1 + N2 + 10	Number of digitizations of 1st contour	
N1 + N2 + 11	Record pointer to 2nd contour	6 words contour
N1 + N2 + 12	Word pointer	
N1 + N2 + 13	Sense of rotation	
N1 + N2 + 14	Nesting	
N1 + N2 + 15	Color	
N1 + N2 + 16	Number of digitizations	
N1 + N2 + 17	Record pointer to 3rd contour	
.	.	
.	.	
.	.	
N1 + N2 + N3 + 2	Number of digitizations of last contour of first character	
	Image information	
N1+N2+N3	Length of data	M=N6×2048+N7
+3	in records (N6)	
+4	in words (N7)	

Word	Meaning	Comment
+5	Image information of first character of this font (in 16 bit words).	
+6	The actual contents	
+7	depends on the format, see appendices H to R.	
.	.	
.	.	
.	.	
N1 + N2 + N3 + M + 2	Final word of image information of first character of this font	
...+3	Length of 2nd character in records	
.	.	
.	.	
.	.	
xyz	Last word of image information of last character of this font	See font header

Description of the IK format

1. Structure of an IK font

The general structure of an IK typeface font is described in Appendix G.

2. Description of the IK format

The IKARUS (IK) format is a digital description of contours of any arbitrary two dimensional plane figure (surface). Such figures are, for example, letters (type characters), company symbols, logos or line graphics.

Contours of these figures are known in mathematics as simple closed curves, continuous everywhere and differentiable almost everywhere. Another description would be:

Closed:	Starting and ending points of a contour are identical.
Simple:	A contour does not cross itself (and in addition, as can easily be imagined, various contours of a figure do not cross each other).
Continuous overall:	A contour has no «gaps».
Differentiable almost overall:	An edge has only a finite number of «corners», it is thus «smooth» almost everywhere.

Each character has its own origin.

Each character has its own coordinate system, with X origin at the X minimum of the character and with the Y origin on the base line.

The IK format describes general curved lines with remarkably little effort, i.e. in a space-saving manner. A contour is described by a sequence of points selected along the contour, each having an (X,Y) coordinate pair referring to a right-angled

(Cartesian) coordinate system, plus one of three point types. This triple designation is referred to here as a «digitization.»

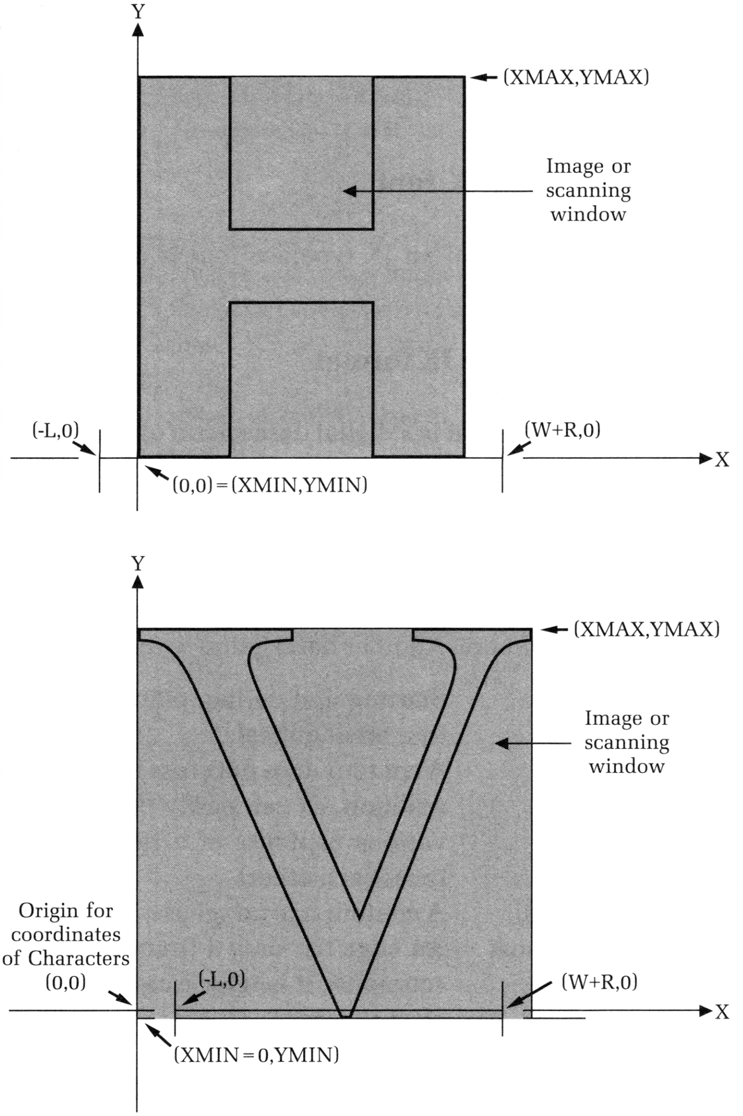

Y
(XMAX,YMAX)
Image or scanning window
(-L,0)
(W+R,0)
X
(0,0) = (XMIN,YMIN)
Y
(XMAX,YMAX)
Image or scanning window
Origin for coordinates of Characters (0,0)
(-L,0)
(W+R,0)
X
(XMIN = 0,YMIN)

Fig. 179
Coordinate system for characters

The meaning of the three different point types will be explained, and we will see why only three different sorts of digitization are necessary and sufficient to comprehensively describe the above named contours.

A contour may be pictured as a succession of straight and curved contour segments or, more succinctly, as a sequence of straights and curves. Two neighbouring straights or two neighbouring curves always, by definition, form a «corner» where they meet. That point is also the first kind of digitization; the corner point. When forming a smooth transition into one another, they are described as a single straight or a single curve.

Digitization points of the second kind are found on curves; called «curve» points. These have the function of «aid points» and are placed whenever the direction of a curve changes by approximately 30 degrees. In addition, all local extrema in X and Y of contours are digitized with curve points. Thus, for example, on a circle these extrema would be the outermost left, right, upper and lower points.

The third digitization is known as the «tangent point». Tangent points are placed wherever a straight runs «tangentially» into a curve or vice versa. At a tangent point straights or curves running in from the left or right thus have the same slope.

To identify beginnings of contours within a figure the IK format has a fourth sort of digitization, the «start» point. To «properly» identify a start point, that point is digitized a second time at the end of a contour with its normal point type.

The point types are coded using the sign of the X and Y coordinates. This limits the absolute values of the coordinates to 32,767. Following table shows how the coordinate values are changed to get a definite coding scheme.

There are only four types of point in the IK format.

Point type	Internal identifier	Change of X coordinate	Change of Y coordinate
Start	12	–(X+1)	Y-YMIN+1
Corner	13	–(X+1)	–(Y-YMIN+1)
Curve	14	X+1	Y-YMIN+1
Tangent	15	X+1	–(Y-YMIN+1)

The X and Y coordinates are stored in the smallest possible units, as a rule 1/100 mm (see also measurement unit in appendix G, character record).

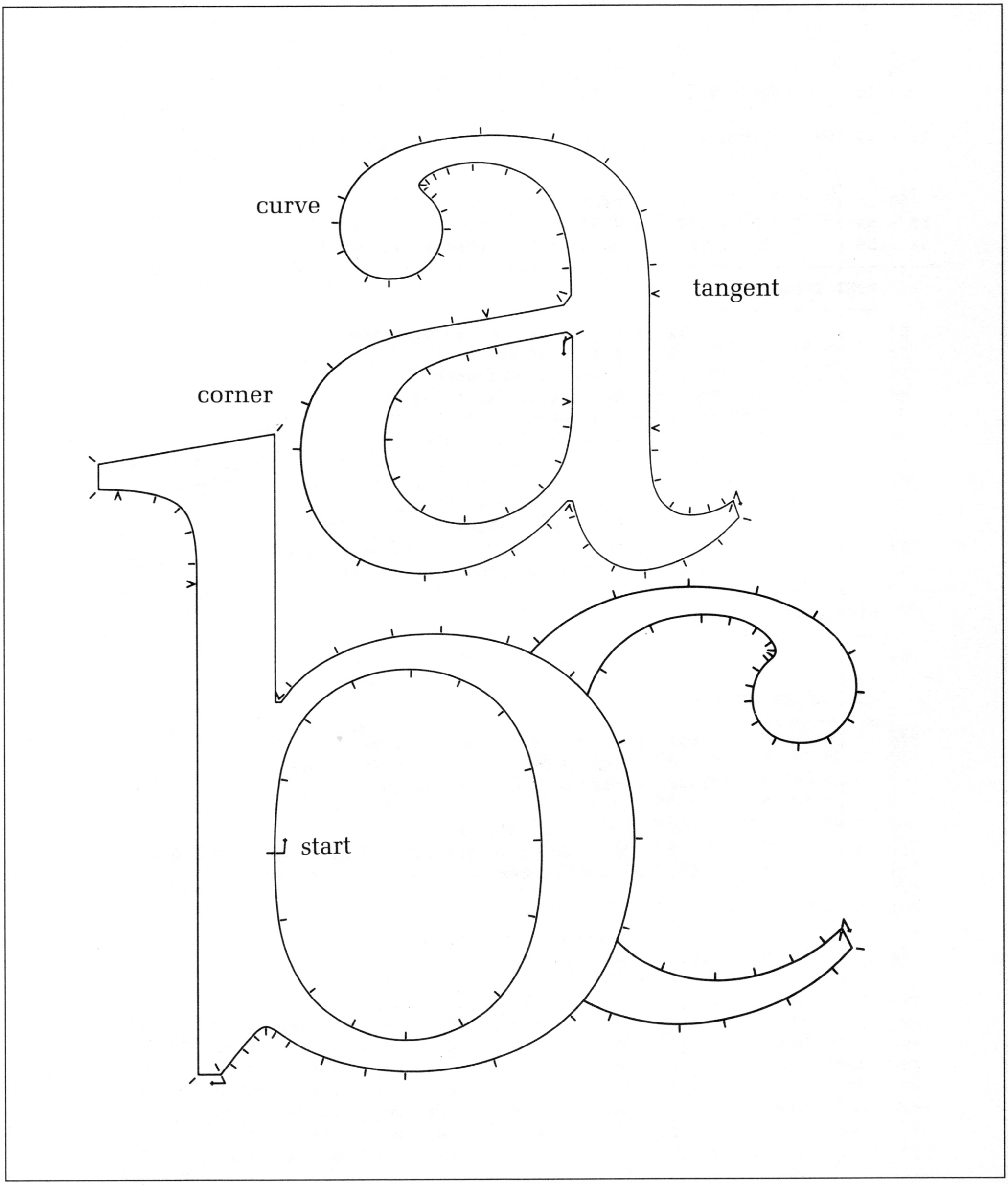
In the URW archives all type fonts are stored with a standard cap height of 100 mm. This is achieved by either adjusting the original artwork to this value using physical reproduction prior to digitization or, where possible, with the help of software after digitization. Size details in IK format headers are assumed to be measured in the said units.

One IK point is counted as one digitization.

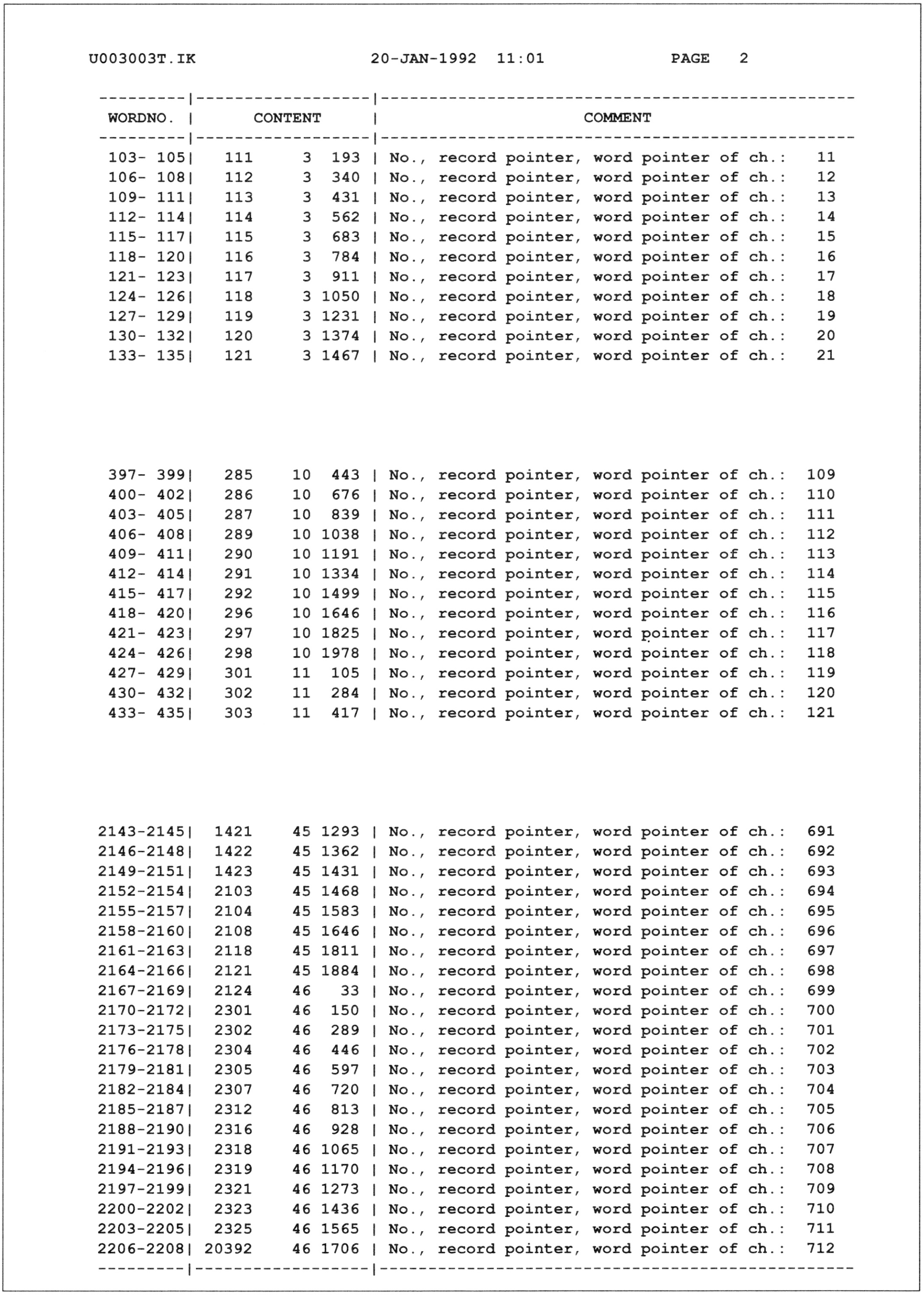
3. Example of an IK format

First we represent lower case letters «a», «b», «c» of the typeface «URW Antiqua 2015 normal», with marks. Subsequently the complete font format is reproduced as a data list.

Fig. 180
IK format representation



```
    U003003T.IK                  20-JAN-1992  11:01            PAGE   1

  ---------|-------------------|---------------------------------------------------
   WORDNO. |      CONTENT      |                    COMMENT
  ---------|-------------------|---------------------------------------------------
      1    |               2208 | Length of font header
  ---------|-------------------|---------------------------------------------------
        NAME SECTION
        ------------
      2    |                 55 | Length of name section
      3    |                  0 | Number on URW list
    4 - 10 |  U003003T.IK       | File name (ASCII)
           |                   |
   11 - 49 |URW Antiqua 2015 Regular            | Font name (ASCII)
           |
           |                   |
     50    |                 IK | Data format (ASCII)
   51 - 53 |      20  1 1992   | Production date:     20-Jan-1992
   54 - 56 |      20  1 1992   | Date of last change: 20-Jan-1992
  ---------|-------------------|---------------------------------------------------
        FONT INFORMATION
        ----------------
     57    |                 12 | Length of font information
     58    |                  1 | Indicator for font
     59    |                712 | Number of characters
     60    |              10000 | Cap height (in 1/100mm)
     61    |              15000 | Body size (in 1/100mm)
     62    |               7000 | x-height (in 1/100mm)
     63    |               3250 | Distance base line - lower body line(in 1/100mm)
     64    |              15000 | Text line distance for setting (in 1/100mm)
     65    |               1300 | Stem thickness (in 1/100mm)
     66    |                  0 | Angle of italisation (in 1/10 degree)
     67    |                 12 | Optimum point size for setting (in p)
     68    |               8462 | Average character width (in 1/100mm)
  ---------|-------------------|---------------------------------------------------
        HIERARCHY SECTION
        -----------------
     69    |                  1 | No hierarchy section in this format
  ---------|-------------------|---------------------------------------------------
         CHARACTER INDEX
        ----------------
     70    |               2139 | Length of character index
     71    |                 46 | LENGTH OF FONT IN PHYSICAL RECORDS
     72    |               1734 | Last word of last record
           |                   |
     73    |                101 | Character number:  1
     74    |                  2 | Pointer to record containing start of character
     75    |               1025 | Word pointer to start of character
           |                   |
     76    |                102 | Ch. no.: 2
     77    |                  2 | Record pointer
     78    |               1140 | Word pointer
           |                   |
    79-  81|   103      2 1303 | No., record pointer, word pointer of ch.:    3
    82-  84|   104      2 1412 | No., record pointer, word pointer of ch.:    4
    85-  87|   105      2 1525 | No., record pointer, word pointer of ch.:    5
    88-  90|   106      2 1652 | No., record pointer, word pointer of ch.:    6
    91-  93|   107      2 1773 | No., record pointer, word pointer of ch.:    7
    94-  96|   108      2 1920 | No., record pointer, word pointer of ch.:    8
    97-  99|   109      3   21 | No., record pointer, word pointer of ch.:    9
   100- 102|   110      3  106 | No., record pointer, word pointer of ch.:   10
  ---------|-------------------|---------------------------------------------------
```



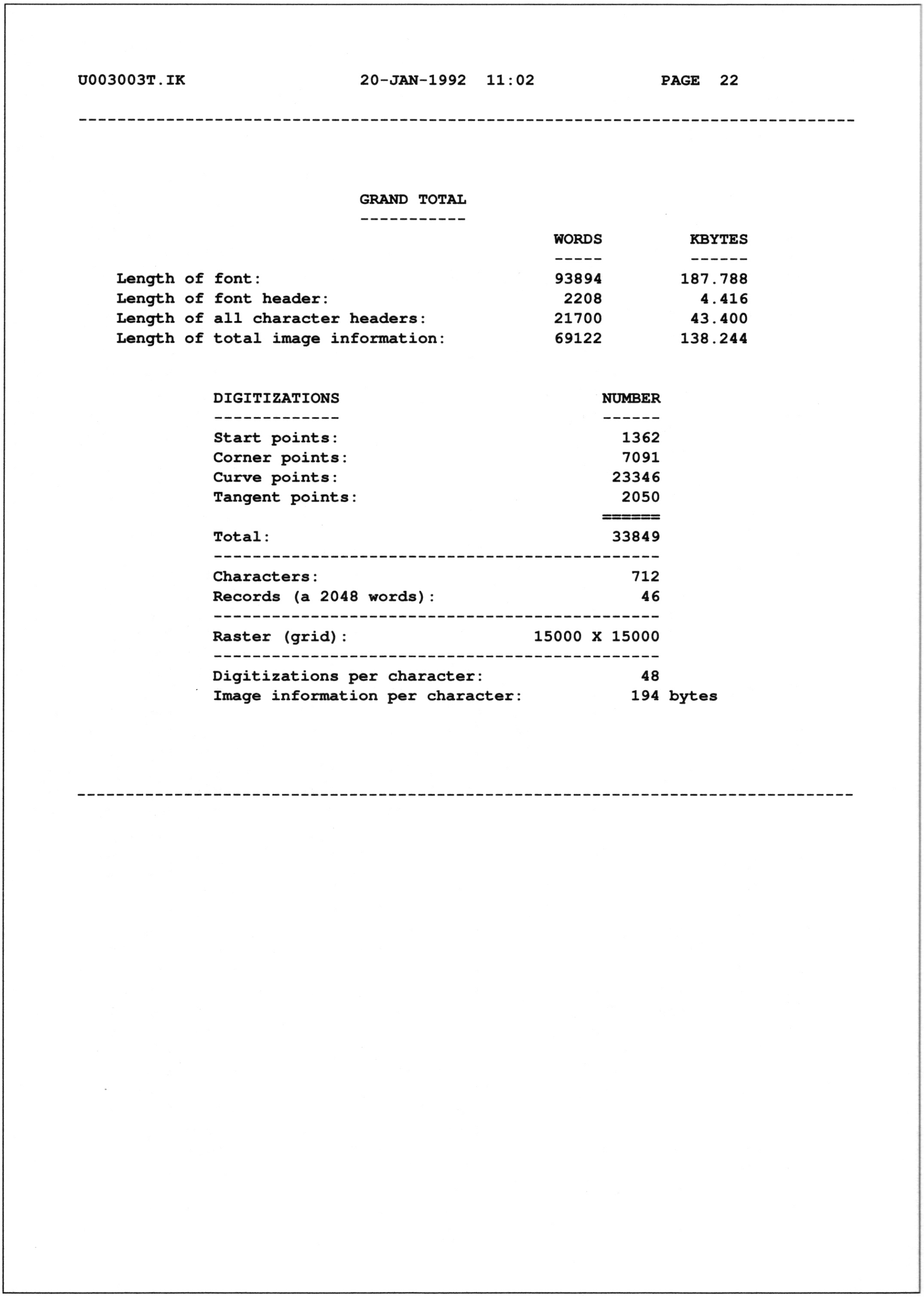
```
U003003T.IK                   20-JAN-1992  11:01            PAGE   2

 ---------|------------------|---------------------------------------------
  WORDNO. |     CONTENT      |                     COMMENT
 ---------|------------------|---------------------------------------------
  103- 105|   111      3  193 | No., record pointer, word pointer of ch.:   11
  106- 108|   112      3  340 | No., record pointer, word pointer of ch.:   12
  109- 111|   113      3  431 | No., record pointer, word pointer of ch.:   13
  112- 114|   114      3  562 | No., record pointer, word pointer of ch.:   14
  115- 117|   115      3  683 | No., record pointer, word pointer of ch.:   15
  118- 120|   116      3  784 | No., record pointer, word pointer of ch.:   16
  121- 123|   117      3  911 | No., record pointer, word pointer of ch.:   17
  124- 126|   118      3 1050 | No., record pointer, word pointer of ch.:   18
  127- 129|   119      3 1231 | No., record pointer, word pointer of ch.:   19
  130- 132|   120      3 1374 | No., record pointer, word pointer of ch.:   20
  133- 135|   121      3 1467 | No., record pointer, word pointer of ch.:   21




  397- 399|   285     10  443 | No., record pointer, word pointer of ch.:  109
  400- 402|   286     10  676 | No., record pointer, word pointer of ch.:  110
  403- 405|   287     10  839 | No., record pointer, word pointer of ch.:  111
  406- 408|   289     10 1038 | No., record pointer, word pointer of ch.:  112
  409- 411|   290     10 1191 | No., record pointer, word pointer of ch.:  113
  412- 414|   291     10 1334 | No., record pointer, word pointer of ch.:  114
  415- 417|   292     10 1499 | No., record pointer, word pointer of ch.:  115
  418- 420|   296     10 1646 | No., record pointer, word pointer of ch.:  116
  421- 423|   297     10 1825 | No., record pointer, word pointer of ch.:  117
  424- 426|   298     10 1978 | No., record pointer, word pointer of ch.:  118
  427- 429|   301     11  105 | No., record pointer, word pointer of ch.:  119
  430- 432|   302     11  284 | No., record pointer, word pointer of ch.:  120
  433- 435|   303     11  417 | No., record pointer, word pointer of ch.:  121




 2143-2145|  1421     45 1293 | No., record pointer, word pointer of ch.:  691
 2146-2148|  1422     45 1362 | No., record pointer, word pointer of ch.:  692
 2149-2151|  1423     45 1431 | No., record pointer, word pointer of ch.:  693
 2152-2154|  2103     45 1468 | No., record pointer, word pointer of ch.:  694
 2155-2157|  2104     45 1583 | No., record pointer, word pointer of ch.:  695
 2158-2160|  2108     45 1646 | No., record pointer, word pointer of ch.:  696
 2161-2163|  2118     45 1811 | No., record pointer, word pointer of ch.:  697
 2164-2166|  2121     45 1884 | No., record pointer, word pointer of ch.:  698
 2167-2169|  2124     46   33 | No., record pointer, word pointer of ch.:  699
 2170-2172|  2301     46  150 | No., record pointer, word pointer of ch.:  700
 2173-2175|  2302     46  289 | No., record pointer, word pointer of ch.:  701
 2176-2178|  2304     46  446 | No., record pointer, word pointer of ch.:  702
 2179-2181|  2305     46  597 | No., record pointer, word pointer of ch.:  703
 2182-2184|  2307     46  720 | No., record pointer, word pointer of ch.:  704
 2185-2187|  2312     46  813 | No., record pointer, word pointer of ch.:  705
 2188-2190|  2316     46  928 | No., record pointer, word pointer of ch.:  706
 2191-2193|  2318     46 1065 | No., record pointer, word pointer of ch.:  707
 2194-2196|  2319     46 1170 | No., record pointer, word pointer of ch.:  708
 2197-2199|  2321     46 1273 | No., record pointer, word pointer of ch.:  709
 2200-2202|  2323     46 1436 | No., record pointer, word pointer of ch.:  710
 2203-2205|  2325     46 1565 | No., record pointer, word pointer of ch.:  711
 2206-2208| 20392     46 1706 | No., record pointer, word pointer of ch.:  712
 ---------|------------------|---------------------------------------------
```

```
U003003T.IK                  20-JAN-1992  11:01            PAGE  15

 ---------|------------------|-------------------------------------------------
 WORDNO.  |     CONTENT      |                     COMMENT
 ---------|------------------|-------------------------------------------------
     1    |                 0| Records      length of character
     2    |               179| Words
 ---------|------------------|-------------------------------------------------
       NAME SECTION
       ------------
     3    |                 3| Length of name section
     4    |               301| Character number
     5    |                 0| Number of following part
 ---------|------------------|-------------------------------------------------
       SETTING INFORMATION
       -------------------
     6    |                12| Length of setting information
     7    |                 1| Character type = letter
     8    |                73| Number of digitizations
     9    |              8240| Total set width (T=L+W+R)
    10    |               799| Left side bearing (L)
    11    |              7291| Width (W)
    12    |               150| Right side bearing (R)
    13    |                 0| X-minimum
    14    |              7291| X-maximum
    15    |              -150| Y-minimum
    16    |              7153| Y-maximum
    17    |                 1| Unit is 1/100mm
 ---------|------------------|-------------------------------------------------
       CONTOUR INDEX
       -------------
    18    |                 0| Records     length of contour index
    19    |                14| Words
          |                  |
    20    |                 0| Record pointer for   1. contour:     0
    21    |                 3| Word pointer                    :    34
    22    |                -1| Sense of rotation of contour = clockwise
    23    |                 0| Nesting: outer  1. order
    24    |                 0| Colour inside: transparent
    25    |                58| Number of digitizations
          |                  |
    26    |                 0| Record pointer for   2. contour:     0
    27    |               119| Word pointer                    :   150
    28    |                -1| Sense of rotation of contour = clockwise
    29    |                 1| Nesting: inner  1. order
    30    |                 0| Colour inside: transparent
    31    |                15| Number of digitizations
          |                  |
 ---------|------------------|-------------------------------------------------
       IMAGE INFORMATION
       -----------------
    32    |                 0| Records     length of image information
    33    |               148| Words
          |                  |
  34-  35|  -7120   1184    | 12   7119   1033 (K,X,Y)      1.DIG; typ: STRP
  36-  37|  -7292   -941    | 13   7291    790 (K,X,Y)      2.DIG; typ: -EP-
  38-  39|   6922    598    | 14   6921    447 (K,X,Y)      3.DIG; typ: .C..
  40-  41|   6347    223    | 14   6346     72 (K,X,Y)      4.DIG; typ: .C..
  42-  43|   5648     49    | 14   5647   -102 (K,X,Y)      5.DIG; typ: .C..
  44-  45|   5211    147    | 14   5210     -4 (K,X,Y)      6.DIG; typ: .C..
  46-  47|   4788    493    | 14   4787    342 (K,X,Y)      7.DIG; typ: .C..
 ---------|------------------|-------------------------------------------------
```

```
U003003T.IK                    20-JAN-1992  11:01            PAGE  16

---------|------------------|----------------------------------------------
 WORDNO. |     CONTENT      |                    COMMENT
---------|------------------|----------------------------------------------
  48-  49|  4572     952    | 14   4571    801  (K,X,Y)      8.DIG; typ: .C..
  50-  51| -4486   -1284    | 13   4485   1133  (K,X,Y)      9.DIG; typ: -EP-
  52-  53|  4067     842    | 14   4066    691  (K,X,Y)     10.DIG; typ: .C..
  54-  55|  3486     406    | 14   3485    255  (K,X,Y)     11.DIG; typ: .C..
  56-  57|  2723      88    | 14   2722    -63  (K,X,Y)     12.DIG; typ: .C..
  58-  59|  2035       1    | 14   2034   -150  (K,X,Y)     13.DIG; typ: .C..
  60-  61|   992     239    | 14    991     88  (K,X,Y)     14.DIG; typ: .C..
  62-  63|   197    1068    | 14    196    917  (K,X,Y)     15.DIG; typ: .C..
  64-  65|     1    2024    | 14      0   1873  (K,X,Y)     16.DIG; typ: .C..
  66-  67|   139    2820    | 14    138   2669  (K,X,Y)     17.DIG; typ: .C..
  68-  69|   543    3417    | 14    542   3266  (K,X,Y)     18.DIG; typ: .C..
  70-  71|  1531    3940    | 14   1530   3789  (K,X,Y)     19.DIG; typ: .C..
  72-  73|  2524    4147    | 14   2523   3996  (K,X,Y)     20.DIG; typ: .C..
  74-  75|  3095   -4245    | 15   3094   4094  (K,X,Y)     21.DIG; typ: >TP<
  76-  77| -4492   -4497    | 13   4491   4346  (K,X,Y)     22.DIG; typ: -EP-
  78-  79|  4477    5101    | 14   4476   4950  (K,X,Y)     23.DIG; typ: .C..
  80-  81|  4415    5714    | 14   4414   5563  (K,X,Y)     24.DIG; typ: .C..
  82-  83|  4070    6457    | 14   4069   6306  (K,X,Y)     25.DIG; typ: .C..
  84-  85|  3468    6775    | 14   3467   6624  (K,X,Y)     26.DIG; typ: .C..
  86-  87|  2860    6849    | 14   2859   6698  (K,X,Y)     27.DIG; typ: .C..
  88-  89|  2177    6748    | 14   2176   6597  (K,X,Y)     28.DIG; typ: .C..
  90-  91|  1941    6635    | 14   1940   6484  (K,X,Y)     29.DIG; typ: .C..
  92-  93|  1801    6525    | 14   1800   6374  (K,X,Y)     30.DIG; typ: .C..
  94-  95|  1722    6391    | 14   1721   6240  (K,X,Y)     31.DIG; typ: .C..
  96-  97|  1776    6284    | 14   1775   6133  (K,X,Y)     32.DIG; typ: .C..
  98-  99|  1906    6157    | 14   1905   6006  (K,X,Y)     33.DIG; typ: .C..
 100- 101|  2053    5931    | 14   2052   5780  (K,X,Y)     34.DIG; typ: .C..
 102- 103|  2104    5655    | 14   2103   5504  (K,X,Y)     35.DIG; typ: .C..
 104- 105|  2007    5263    | 14   2006   5112  (K,X,Y)     36.DIG; typ: .C..
 106- 107|  1682    4930    | 14   1681   4779  (K,X,Y)     37.DIG; typ: .C..
 108- 109|  1237    4828    | 14   1236   4677  (K,X,Y)     38.DIG; typ: .C..
 110- 111|   874    4899    | 14    873   4748  (K,X,Y)     39.DIG; typ: .C..
 112- 113|   486    5304    | 14    485   5153  (K,X,Y)     40.DIG; typ: .C..
 114- 115|   403    5705    | 14    402   5554  (K,X,Y)     41.DIG; typ: .C..
 116- 117|   499    6160    | 14    498   6009  (K,X,Y)     42.DIG; typ: .C..
 118- 119|   963    6751    | 14    962   6600  (K,X,Y)     43.DIG; typ: .C..
 120- 121|  1902    7168    | 14   1901   7017  (K,X,Y)     44.DIG; typ: .C..
 122- 123|  3163    7304    | 14   3162   7153  (K,X,Y)     45.DIG; typ: .C..
 124- 125|  4149    7213    | 14   4148   7062  (K,X,Y)     46.DIG; typ: .C..
 126- 127|  5038    6835    | 14   5037   6684  (K,X,Y)     47.DIG; typ: .C..
 128- 129|  5647    6032    | 14   5646   5881  (K,X,Y)     48.DIG; typ: .C..
 130- 131|  5808    5150    | 14   5807   4999  (K,X,Y)     49.DIG; typ: .C..
 132- 133|  5814   -4677    | 15   5813   4526  (K,X,Y)     50.DIG; typ: >TP<
 134- 135|  5814   -2222    | 15   5813   2071  (K,X,Y)     51.DIG; typ: >TP<
 136- 137|  5820    1812    | 14   5819   1661  (K,X,Y)     52.DIG; typ: .C..
 138- 139|  5870    1344    | 14   5869   1193  (K,X,Y)     53.DIG; typ: .C..
 140- 141|  6072     958    | 14   6071    807  (K,X,Y)     54.DIG; typ: .C..
 142- 143|  6394     841    | 14   6393    690  (K,X,Y)     55.DIG; typ: .C..
 144- 145|  6748     938    | 14   6747    787  (K,X,Y)     56.DIG; typ: .C..
 146- 147|  6972    1080    | 14   6971    929  (K,X,Y)     57.DIG; typ: .C..
 148- 149| -7120   -1184    | 13   7119   1033  (K,X,Y)     58.DIG; typ: -EP-
 150- 151| -4514    4035    | 12   4513   3884  (K,X,Y)     59.DIG; typ: STRP
 152- 153|  4514   -2866    | 15   4513   2715  (K,X,Y)     60.DIG; typ: >TP<
 154- 155|  4489    2434    | 14   4488   2283  (K,X,Y)     61.DIG; typ: .C..
 156- 157|  4352    1832    | 14   4351   1681  (K,X,Y)     62.DIG; typ: .C..
 158- 159|  4057    1388    | 14   4056   1237  (K,X,Y)     63.DIG; typ: .C..
---------|------------------|----------------------------------------------
```

```
U003003T.IK                  20-JAN-1992  11:01           PAGE  17

 ---------|------------------|------------------------------------------------
  WORDNO. |     CONTENT      |                  COMMENT
 ---------|------------------|------------------------------------------------
  160- 161|   3463     974   | 14   3462     823 (K,X,Y)      64.DIG; typ: .C..
  162- 163|   2737     822   | 14   2736     671 (K,X,Y)      65.DIG; typ: .C..
  164- 165|   1991    1023   | 14   1990     872 (K,X,Y)      66.DIG; typ: .C..
  166- 167|   1522    1552   | 14   1521    1401 (K,X,Y)      67.DIG; typ: .C..
  168- 169|   1390    2187   | 14   1389    2036 (K,X,Y)      68.DIG; typ: .C..
  170- 171|   1556    2933   | 14   1555    2782 (K,X,Y)      69.DIG; typ: .C..
  172- 173|   2077    3437   | 14   2076    3286 (K,X,Y)      70.DIG; typ: .C..
  174- 175|   2752    3675   | 14   2751    3524 (K,X,Y)      71.DIG; typ: .C..
  176- 177|   3227    3790   | 14   3226    3639 (K,X,Y)      72.DIG; typ: .C..
  178- 179|  -4514   -4035   | 13   4513    3884 (K,X,Y)      73.DIG; typ: -EP-
 ---------|------------------|------------------------------------------------



 ---------|------------------|------------------------------------------------
     1    |                 0| Records      length of character
     2    |               133| Words
 ---------|------------------|------------------------------------------------
       NAME SECTION
       ------------
     3    |                 3| Length of name section
     4    |               302| Character number
     5    |                 0| Number of following part
 ---------|------------------|------------------------------------------------
       SETTING INFORMATION
       -------------------
     6    |                12| Length of setting information
     7    |                 1| Character type = letter
     8    |                50| Number of digitizations
     9    |              9413| Total set width (T=L+W+R)
    10    |               -40| Left side bearing (L)
    11    |              8781| Width (W)
    12    |               672| Right side bearing (R)
    13    |                 0| X-minimum
    14    |              8781| X-maximum
    15    |              -200| Y-minimum
    16    |             10350| Y-maximum
    17    |                 1| Unit is 1/100mm
 ---------|------------------|------------------------------------------------
       CONTOUR INDEX
       -------------
    18    |                 0| Records     length of contour index
    19    |                14| Words
          |                  |
    20    |                 0| Record pointer for   1. contour:      0
    21    |                 3| Word pointer                     :    34
    22    |                -1| Sense of rotation of contour = clockwise
    23    |                 0| Nesting: outer  1. order
    24    |                 0| Colour inside: transparent
    25    |                33| Number of digitizations
          |                  |
    26    |                 0| Record pointer for   2. contour:      0
    27    |                69| Word pointer                     :   100
    28    |                -1| Sense of rotation of contour = clockwise
    29    |                 1| Nesting: inner  1. order
    30    |                 0| Colour inside: transparent
    31    |                17| Number of digitizations
 ---------|------------------|------------------------------------------------
```

```
U003003T.IK                  20-JAN-1992  11:01            PAGE  19

---------|------------------|------------------------------------------------
 WORDNO. |      CONTENT     |                    COMMENT
---------|------------------|------------------------------------------------
      IMAGE INFORMATION
      -----------------
   32    |                0 | Records     length of image information
   33    |              102 | Words
         |                  |
  34-  35|  -1905       1   | 12    1904    -200 (K,X,Y)       1.DIG; typ: STRP
  36-  37|  -1529      -1   | 13    1528    -200 (K,X,Y)       2.DIG; typ: -EP-
  38-  39|   1529   -8197   | 15    1528    7996 (K,X,Y)       3.DIG; typ: >TP<
  40-  41|   1527    8536   | 14    1526    8335 (K,X,Y)       4.DIG; typ: .C..
  42-  43|   1471    9067   | 14    1470    8866 (K,X,Y)       5.DIG; typ: .C..
  44-  45|   1263    9479   | 14    1262    9278 (K,X,Y)       6.DIG; typ: .C..
  46-  47|    854    9679   | 14     853    9478 (K,X,Y)       7.DIG; typ: .C..
  48-  49|    226   -9762   | 15     225    9561 (K,X,Y)       8.DIG; typ: >TP<
  50-  51|     -1   -9771   | 13       0    9570 (K,X,Y)       9.DIG; typ: -EP-
  52-  53|     -1  -10201   | 13       0   10000 (K,X,Y)      10.DIG; typ: -EP-
  54-  55|  -2829  -10551   | 13    2828   10350 (K,X,Y)      11.DIG; typ: -EP-
  56-  57|  -2829   -6107   | 13    2828    5906 (K,X,Y)      12.DIG; typ: -EP-
  58-  59|   3029    6356   | 14    3028    6155 (K,X,Y)      13.DIG; typ: .C..
  60-  61|   3690    6936   | 14    3689    6735 (K,X,Y)      14.DIG; typ: .C..
  62-  63|   4653    7300   | 14    4652    7099 (K,X,Y)      15.DIG; typ: .C..
  64-  65|   5312    7356   | 14    5311    7155 (K,X,Y)      16.DIG; typ: .C..
  66-  67|   6591    7154   | 14    6590    6953 (K,X,Y)      17.DIG; typ: .C..
  68-  69|   7626    6579   | 14    7625    6378 (K,X,Y)      18.DIG; typ: .C..
  70-  71|   8425    5516   | 14    8424    5315 (K,X,Y)      19.DIG; typ: .C..
  72-  73|   8782    3738   | 14    8781    3537 (K,X,Y)      20.DIG; typ: .C..
  74-  75|   8603    2436   | 14    8602    2235 (K,X,Y)      21.DIG; typ: .C..
  76-  77|   7718     932   | 14    7717     731 (K,X,Y)      22.DIG; typ: .C..
  78-  79|   6433     218   | 14    6432      17 (K,X,Y)      23.DIG; typ: .C..
  80-  81|   5149      48   | 14    5148    -153 (K,X,Y)      24.DIG; typ: .C..
  82-  83|   4318     114   | 14    4317     -87 (K,X,Y)      25.DIG; typ: .C..
  84-  85|   3619     299   | 14    3618      98 (K,X,Y)      26.DIG; typ: .C..
  86-  87|   3071     573   | 14    3070     372 (K,X,Y)      27.DIG; typ: .C..
  88-  89|   2830     728   | 14    2829     527 (K,X,Y)      28.DIG; typ: .C..
  90-  91|   2662     800   | 14    2661     599 (K,X,Y)      29.DIG; typ: .C..
  92-  93|   2445     664   | 14    2444     463 (K,X,Y)      30.DIG; typ: .C..
  94-  95|   2254     444   | 14    2253     243 (K,X,Y)      31.DIG; typ: .C..
  96-  97|   2053     192   | 14    2052      -9 (K,X,Y)      32.DIG; typ: .C..
  98-  99|  -1905      -1   | 13    1904    -200 (K,X,Y)      33.DIG; typ: -EP-
 100- 101|  -2829    3741   | 12    2828    3540 (K,X,Y)      34.DIG; typ: STRP
 102- 103|   2915    4933   | 14    2914    4732 (K,X,Y)      35.DIG; typ: .C..
 104- 105|   3324    5949   | 14    3323    5748 (K,X,Y)      36.DIG; typ: .C..
 106- 107|   4117    6571   | 14    4116    6370 (K,X,Y)      37.DIG; typ: .C..
 108- 109|   5057    6769   | 14    5056    6568 (K,X,Y)      38.DIG; typ: .C..
 110- 111|   5922    6596   | 14    5921    6395 (K,X,Y)      39.DIG; typ: .C..
 112- 113|   6636    6071   | 14    6635    5870 (K,X,Y)      40.DIG; typ: .C..
 114- 115|   7123    5096   | 14    7122    4895 (K,X,Y)      41.DIG; typ: .C..
 116- 117|   7277    3701   | 14    7276    3500 (K,X,Y)      42.DIG; typ: .C..
 118- 119|   7189    2627   | 14    7188    2426 (K,X,Y)      43.DIG; typ: .C..
 120- 121|   6580    1291   | 14    6579    1090 (K,X,Y)      44.DIG; typ: .C..
 122- 123|   5948     803   | 14    5947     602 (K,X,Y)      45.DIG; typ: .C..
 124- 125|   4956     574   | 14    4955     373 (K,X,Y)      46.DIG; typ: .C..
 126- 127|   4099     745   | 14    4098     544 (K,X,Y)      47.DIG; typ: .C..
 128- 129|   3338    1319   | 14    3337    1118 (K,X,Y)      48.DIG; typ: .C..
 130- 131|   2893    2581   | 14    2892    2380 (K,X,Y)      49.DIG; typ: .C..
 132- 133|   2829    3741   | 14    2828    3540 (K,X,Y)      50.DIG; typ: .C..
---------|------------------|------------------------------------------------
```



```
U003003T.IK                 20-JAN-1992  11:02            PAGE  22
-------------------------------------------------------------------------------


                             GRAND TOTAL
                             -----------
                                                 WORDS         KBYTES
                                                 -----         ------
    Length of font:                              93894         187.788
    Length of font header:                        2208           4.416
    Length of all character headers:             21700          43.400
    Length of total image information:           69122         138.244


              DIGITIZATIONS                           NUMBER
              -------------                           ------
              Start points:                             1362
              Corner points:                            7091
              Curve points:                            23346
              Tangent points:                           2050
                                                      ======
              Total:                                   33849
              ----------------------------------------------
              Characters:                                712
              Records (a 2048 words):                     46
              ----------------------------------------------
              Raster (grid):                   15000 X 15000
              ----------------------------------------------
              Digitizations per character:                48
              Image information per character:           194 bytes



-------------------------------------------------------------------------------
```

Description of the BE, CN and QQ format

1. Structure of the Formats

The **Bézier**, **Conic** and **Quadratic Spline formats** have the same structure as the IK format, the sole difference being the meaning of the digitized points in their image information.

2. Description of the Formats

None of three formats contain instructions.

Unlike the IK format, BE, CN and QQ formats all include in their image information an explicit definition of the parameters for mathematical functions. The IK format defines the coordinates and point types, but not the parameters for mathematical functions. The format is thus not restricted to any particular method of curve representation making it relatively easy to write fast and accurate programs for conversion from IK to the BE, CN and QQ formats as well as others (DI, VC, BS ...). The IK format thus constitutes a further abstraction of the way in which characters can be stored. It is not tied to a specific type of mathematics, making it a most suitable format for manual or automatic digitizing of artwork, but also for being the format for long-term data storage.

The image data of all three formats (BE, CN and QQ) is described by so-called control and anchor points, coded with two words (two bytes each) for X and Y. As in the IK format, the highest-value bits of both words describe the point type. In the BE, CN and QQ formats, the point types also act as parameters for the function generating curves. The X and Y coordinate values must be within the range $X,Y \leq 32{,}767$.

IK format

Point type	Start	Corner	Curve	Tangent
Bit 15 (1st Word Bit 15 (2nd Word)	1 0	1 1	0 0	0 1
Identification	12	13	14	15

BE format

Point type	Moveto (start)	Anchor Lineto	Control-point	Anchor Curveto
Bit 15 (1st Word) Bit 15 (2nd Word)	1 0	1 1	0 0	0 1
Identification	12	13	14	15

CN format

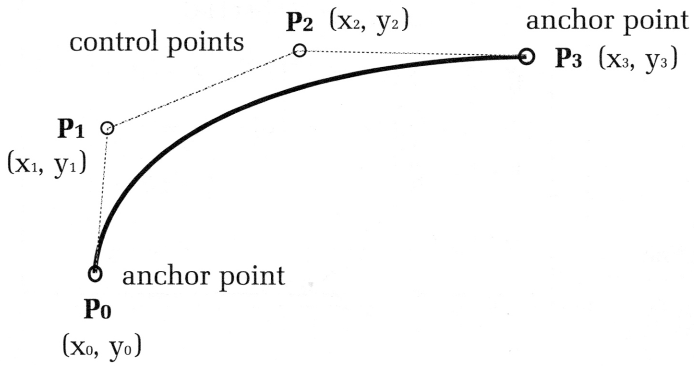
Point type	Moveto (start)	Control-point	Sharp-ness	Anchor-point
Bit 15 (1st Word) Bit 15 (2nd Word)	1 0	1 1	0 0	0 1
Identification	12	13	14	15

QQ format

Point type	Moveto (start)	Anchor Lineto	Control-point	Anchor Curveto
Bit 15 (1st Wort) Bit 15 (2nd Wort)	1 0	1 1	0 0	0 1
Identification	12	13	14	15

3. BE Format

A **Bézier curve** is constructed on the basis of two control points (P1, P2) and two anchor points (P0, P3). The anchor points are located at the beginning and end of the Bézier curve section.


control points
P2 (x2, y2)
anchor point
P3 (x3, y3)
P1
(x1, y1)
anchor point
P0
(x0, y0)

$$x(t) = x_0 \bullet (1-t)^3 + 3x_1 \bullet (1-t)^2 t + 3x_2 \bullet (1-t)t^2 + x_3 \bullet t^3$$

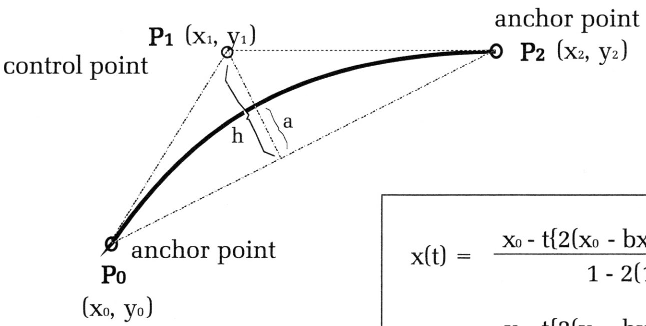
$$y(t) = y_0 \bullet (1-t)^3 + 3y_1 \bullet (1-t)^2 t + 3y_2 \bullet (1-t)t^2 + y_3 \bullet t^3$$

$$0 \leq t \leq 1$$

Fig. 181

4. CN Format

The Conic format is based on conics and straights. These conics are parametrized by a control point (P1), the sharpness (S) and two anchor points (P0, P2). The sharpness is represented as a fraction, Sx the numerator and Sy the denominator. The denominator is usually defined as a fixed 15,000.



P_1 (x_1, y_1)
control point
anchor point
P_2 (x_2, y_2)
h
a
anchor point
P_0
(x_0, y_0)

$$x(t) = \frac{x_0 - t\{2(x_0 - bx_1) - t\,(x_0 - 2bx_1 + x_2)\}}{1 - 2(1-b)\,t\,(1-t)}$$

$$y(t) = \frac{y_0 - t\{2(y_0 - by_1) - t\,(y_0 - 2by_1 + y_2)\}}{1 - 2(1-b)\,t\,(1-t)}$$

$$b = \frac{s}{1-s}$$

$$s = \frac{a}{h}$$

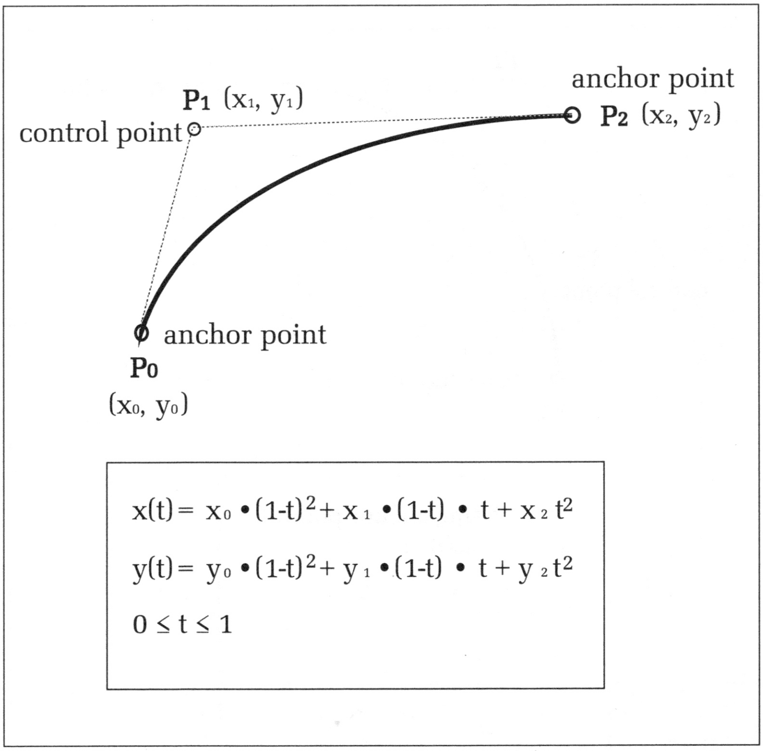
$$0 \le t \le 1$$

$s = 0$	straight line
$s < \frac{1}{2}$	ellipse
$s = \sqrt{2}-1$	circle for symmetrical triangle
$s = \frac{1}{2}$	parabola
$s > \frac{1}{2}$	hyperbola
$s = 1$	2 straight lines

Fig. 182

5. QQ Format

The Quadratic Spline Format contains straights and parabola (quadratic splines). The straights are represented by two anchor points, the parabola by an extra control point located between the two anchor points.


anchor point
P1 (x1, y1)
control point
P2 (x2, y2)
anchor point
P0
(x0, y0)

$$x(t) = x_0 \bullet (1-t)^2 + x_1 \bullet (1-t) \bullet t + x_2 t^2$$

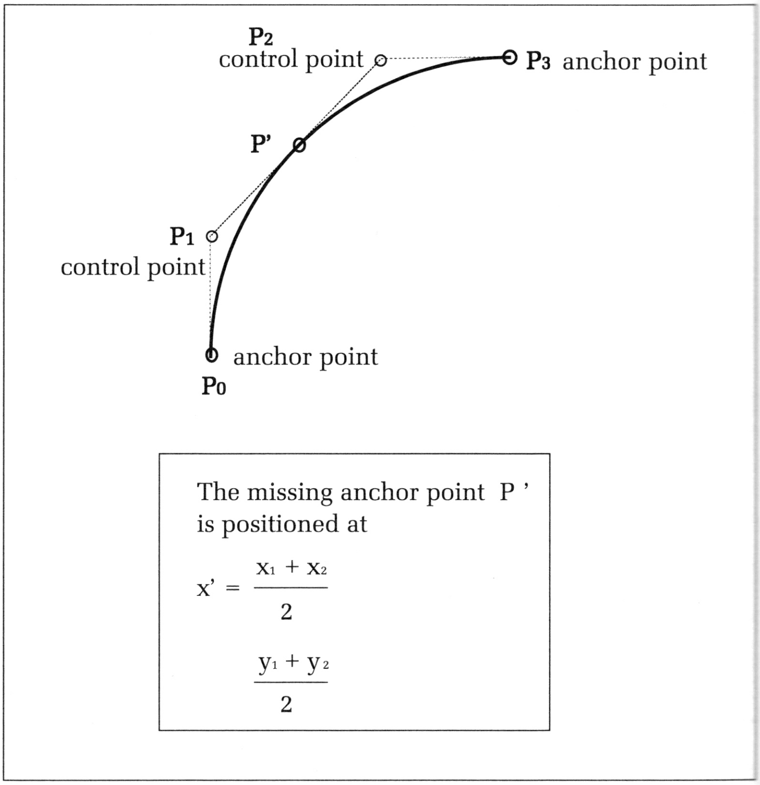
$$y(t) = y_0 \bullet (1-t)^2 + y_1 \bullet (1-t) \bullet t + y_2 t^2$$

$$0 \leq t \leq 1$$

Fig. 183

A Special Case

As a rule, there is a chain of parabola, with an anchor point at either end and a series of control points between the two ends. It is assumed implicitly that there is an anchor point between every two control points, located in the middle of the straight line which would join the two control points:



The missing anchor point P ' is positioned at

$$x' = \frac{x_1 + x_2}{2}$$

$$\frac{y_1 + y_2}{2}$$

Fig. 184

6. Remarks on BE, CN and QQ

Unlike the IK format, these three formats use control points not normally positioned on the contour. These control values are, however, used to compute the bounding box coordinates (Xmin, Xmax, Ymin and Ymax), which in some cases, might create a character width greater or smaller than the difference between Xmax and Xmin. However, as only anchor points have extreme positions, we did not notice these deviations until now.

Sharpness is never handled by shift operations, an important point to be observed by application programs.

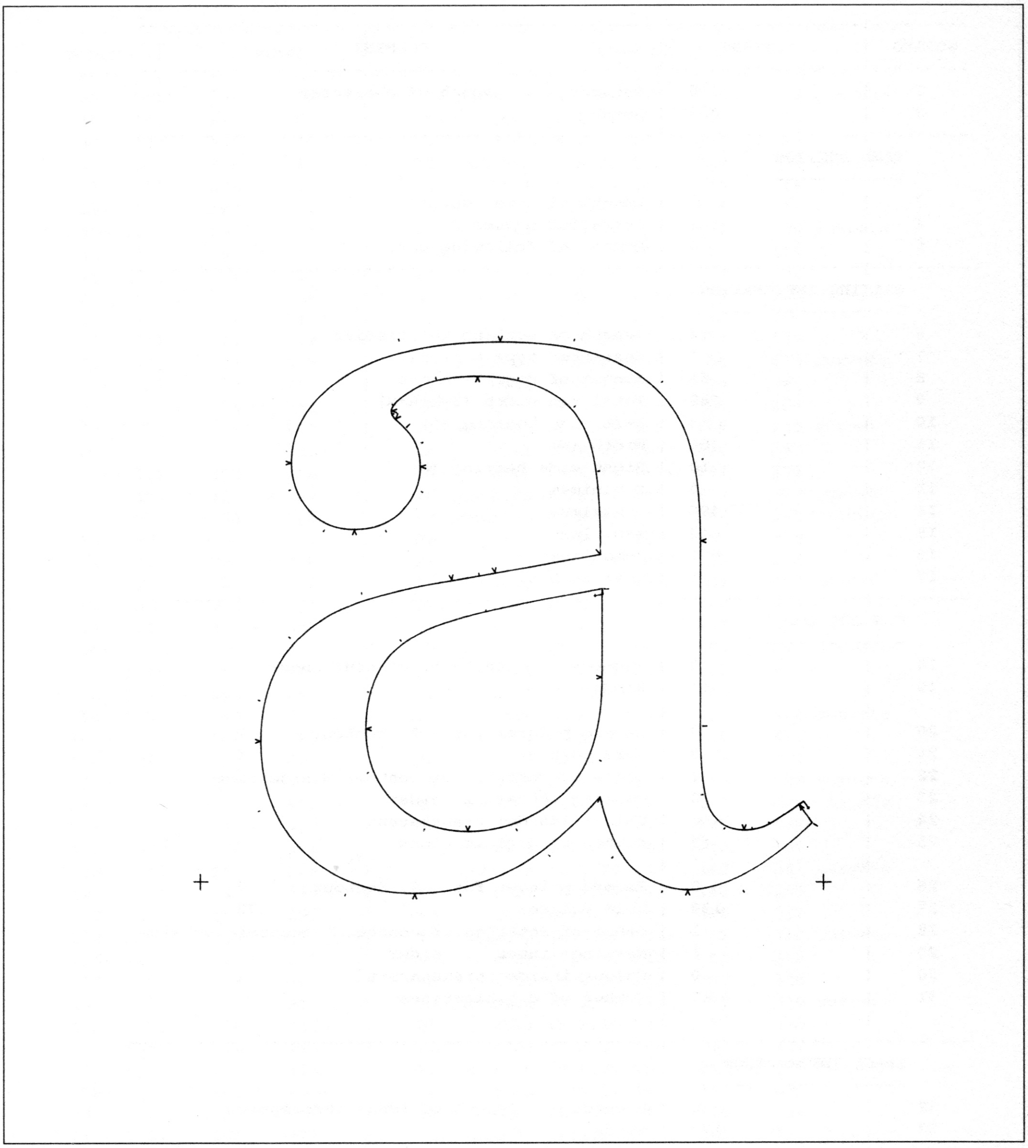


Fig 185
A BE format character, with marks to show point types

```
U003003T.BE                   20-JAN-1992  11:54            PAGE  15

 ---------|-----------------|-------------------------------------------------
  WORDNO. |     CONTENT     |                      COMMENT
 ---------|-----------------|-------------------------------------------------
     1    |                0 | Records      length of character
     2    |              203 | Words
 ---------|-----------------|-------------------------------------------------
       NAME SECTION
       ------------
     3    |                3 | Length of name section
     4    |              301 | Character number
     5    |                0 | Number of following part
 ---------|-----------------|-------------------------------------------------
       SETTING INFORMATION
       -------------------
     6    |               12 | Length of setting information
     7    |                1 | Character type = letter
     8    |               85 | Number of digitizations
     9    |              549 | Total set width (T=L+W+R)
    10    |               53 | Left side bearing (L)
    11    |              486 | Width (W)
    12    |               10 | Right side bearing (R)
    13    |                0 | X-minimum
    14    |              486 | X-maximum
    15    |              -10 | Y-minimum
    16    |              477 | Y-maximum
    17    |                1 | Unit is 1 pixel
 ---------|-----------------|-------------------------------------------------
       CONTOUR INDEX
       -------------
    18    |                0 | Records     length of contour index
    19    |               14 | Words
          |                 |
    20    |                0 | Record pointer for   1. contour:      0
    21    |                3 | Word pointer                   :    34
    22    |               -1 | Sense of rotation of contour = clockwise
    23    |                0 | Nesting: outer  1. order
    24    |                0 | Colour inside: transparent
    25    |               68 | Number of digitizations
          |                 |
    26    |                0 | Record pointer for   2. contour:      0
    27    |              139 | Word pointer                   :   170
    28    |                1 | Sense of rotation of contour = counterclockwise
    29    |                1 | Nesting: inner  1. order
    30    |                0 | Colour inside: transparent
    31    |               17 | Number of digitizations
          |                 |
 ---------|-----------------|-------------------------------------------------
       IMAGE INFORMATION
       -----------------
    32    |                0 | Records     length of image information
    33    |              172 | Words
          |                 |
   34-  35|  -476      80    |     1. startp. (K:X,Y) 12:   475      69  moveto
   36-  37|  -487     -64    |     2. anchorp (K:X,Y) 13:   486      53  lineto
   38-  39|   443      21    |     3.    ctrl (K:X,Y) 14:   442      10
   40-  41|   412       4    |     4.    ctrl (K:X,Y) 14:   411      -7
   42-  43|   377      -4    |     5. anchorp (K:X,Y) 15:   376      -7 curveto
   44-  45|   355       4    |     6.    ctrl (K:X,Y) 14:   354      -7
   46-  47|   332      16    |     7.    ctrl (K:X,Y) 14:   331       5
 ---------|-----------------|-------------------------------------------------
```

```
U003003T.BE                     20-JAN-1992  11:54           PAGE  16

 ---------|------------------|---------------------------------------------
  WORDNO. |     CONTENT      |                     COMMENT
 ---------|------------------|---------------------------------------------
   48-  49|   320     -34    |     8. anchorp (K:X,Y) 15:   319      23 curveto
   50-  51|   311      47    |     9.    ctrl (K:X,Y) 14:   310      36
   52-  53|   307      58    |    10.    ctrl (K:X,Y) 14:   306      47
   54-  55|   300     -87    |    11. anchorp (K:X,Y) 15:   299      76 curveto
   56-  57|   245      25    |    12.    ctrl (K:X,Y) 14:   244      14
   58-  59|   199       1    |    13.    ctrl (K:X,Y) 14:   198     -10
   60-  61|   137      -1    |    14. anchorp (K:X,Y) 15:   136     -10 curveto
   62-  63|    55       1    |    15.    ctrl (K:X,Y) 14:    54     -10
   64-  65|     1      54    |    16.    ctrl (K:X,Y) 14:     0      43
   66-  67|     1    -136    |    17. anchorp (K:X,Y) 15:     0     125 curveto
   68-  69|     1     175    |    18.    ctrl (K:X,Y) 14:     0     164
   70-  71|    14     209    |    19.    ctrl (K:X,Y) 14:    13     198
   72-  73|    37    -229    |    20. anchorp (K:X,Y) 15:    36     218 curveto
   74-  75|    67     255    |    21.    ctrl (K:X,Y) 14:    66     244
   76-  77|    96     265    |    22.    ctrl (K:X,Y) 14:    95     254
   78-  79|   169    -277    |    23. anchorp (K:X,Y) 15:   168     266 curveto
   80-  81|   185     280    |    24.    ctrl (K:X,Y) 14:   184     269
   82-  83|   198     282    |    25.    ctrl (K:X,Y) 14:   197     271
   84-  85|   207    -284    |    26. anchorp (K:X,Y) 15:   206     273 curveto
   86-  87|  -300    -301    |    27. anchorp (K:X,Y) 13:   299     290  lineto
   88-  89|   299     371    |    28.    ctrl (K:X,Y) 14:   298     360
   90-  91|   297     390    |    29.    ctrl (K:X,Y) 14:   296     379
   92-  93|   285    -413    |    30. anchorp (K:X,Y) 15:   284     402 curveto
   94-  95|   271     442    |    31.    ctrl (K:X,Y) 14:   270     431
   96-  97|   238     458    |    32.    ctrl (K:X,Y) 14:   237     447
   98-  99|   192    -458    |    33. anchorp (K:X,Y) 15:   191     447 curveto
  100- 101|   154     458    |    34.    ctrl (K:X,Y) 14:   153     447
  102- 103|   116     442    |    35.    ctrl (K:X,Y) 14:   115     431
  104- 105|   116    -427    |    36. anchorp (K:X,Y) 15:   115     416 curveto
  106- 107|   116     425    |    37.    ctrl (K:X,Y) 14:   115     414
  108- 109|   117     422    |    38.    ctrl (K:X,Y) 14:   116     411
  110- 111|   119    -420    |    39. anchorp (K:X,Y) 15:   118     409 curveto
  112- 113|  -128    -411    |    40. anchorp (K:X,Y) 13:   127     400  lineto
  114- 115|   136     403    |    41.    ctrl (K:X,Y) 14:   135     392
  116- 117|   141     391    |    42.    ctrl (K:X,Y) 14:   140     380
  118- 119|   141    -378    |    43. anchorp (K:X,Y) 15:   140     367 curveto
  120- 121|   141     346    |    44.    ctrl (K:X,Y) 14:   140     335
  122- 123|   117     323    |    45.    ctrl (K:X,Y) 14:   116     312
  124- 125|    83    -323    |    46. anchorp (K:X,Y) 15:    82     312 curveto
  126- 127|    51     323    |    47.    ctrl (K:X,Y) 14:    50     312
  128- 129|    28     347    |    48.    ctrl (K:X,Y) 14:    27     336
  130- 131|    28    -381    |    49. anchorp (K:X,Y) 15:    27     370 curveto
  132- 133|    28     446    |    50.    ctrl (K:X,Y) 14:    27     435
  134- 135|    99     488    |    51.    ctrl (K:X,Y) 14:    98     477
  136- 137|   212    -488    |    52. anchorp (K:X,Y) 15:   211     477 curveto
  138- 139|   288     488    |    53.    ctrl (K:X,Y) 14:   287     477
  140- 141|   337     469    |    54.    ctrl (K:X,Y) 14:   336     458
  142- 143|   365    -427    |    55. anchorp (K:X,Y) 15:   364     416 curveto
  144- 145|   383     401    |    56.    ctrl (K:X,Y) 14:   382     390
  146- 147|   389     374    |    57.    ctrl (K:X,Y) 14:   388     363
  148- 149|   389    -313    |    58. anchorp (K:X,Y) 15:   388     302 curveto
  150- 151|  -389    -149    |    59. anchorp (K:X,Y) 13:   388     138  lineto
  152- 153|   389     119    |    60.    ctrl (K:X,Y) 14:   388     108
  154- 155|   389     105    |    61.    ctrl (K:X,Y) 14:   388      94
  156- 157|   392     -91    |    62. anchorp (K:X,Y) 15:   391      80 curveto
  158- 159|   396      70    |    63.    ctrl (K:X,Y) 14:   395      59
 ---------|------------------|---------------------------------------------
```

```
U003003T.BE                    20-JAN-1992  11:54           PAGE  17

 ---------|------------------|--------------------------------------------------
  WORDNO. |      CONTENT     |                      COMMENT
 ---------|------------------|--------------------------------------------------
  160- 161|   409      57    |    64.     ctrl  (K:X,Y) 14:   408     46
  162- 163|   426     -57    |    65. anchorp  (K:X,Y) 15:   425     46 curveto
  164- 165|   441      57    |    66.     ctrl  (K:X,Y) 14:   440     46
  166- 167|   447      60    |    67.     ctrl  (K:X,Y) 14:   446     49
  168- 169|   476     -80    |    68. anchorp  (K:X,Y) 15:   475     69 curveto
  170- 171|  -302     270    |    69. startp.  (K:X,Y) 12:   301    259  moveto
  172- 173|   227     257    |    70.     ctrl  (K:X,Y) 14:   226    246
  174- 175|   184     247    |    71.     ctrl  (K:X,Y) 14:   183    236
  176- 177|   157    -238    |    72. anchorp  (K:X,Y) 15:   156    227 curveto
  178- 179|   113     222    |    73.     ctrl  (K:X,Y) 14:   112    211
  180- 181|    94     194    |    74.     ctrl  (K:X,Y) 14:    93    183
  182- 183|    94    -147    |    75. anchorp  (K:X,Y) 15:    93    136 curveto
  184- 185|    94      93    |    76.     ctrl  (K:X,Y) 14:    93     82
  186- 187|   131      56    |    77.     ctrl  (K:X,Y) 14:   130     45
  188- 189|   183     -56    |    78. anchorp  (K:X,Y) 15:   182     45 curveto
  190- 191|   224      56    |    79.     ctrl  (K:X,Y) 14:   223     45
  192- 193|   264      77    |    80.     ctrl  (K:X,Y) 14:   263     66
  194- 195|   285    -110    |    81. anchorp  (K:X,Y) 15:   284     99 curveto
  196- 197|   295     128    |    82.     ctrl  (K:X,Y) 14:   294    117
  198- 199|   302     158    |    83.     ctrl  (K:X,Y) 14:   301    147
  200- 201|   302    -192    |    84. anchorp  (K:X,Y) 15:   301    181 curveto
  202- 203|  -302    -270    |    85. anchorp  (K:X,Y) 13:   301    259  lineto
 ---------|------------------|--------------------------------------------------


--------------------------------------------------------------------------------


                            GRAND TOTAL
                            -----------
                                                   WORDS         KBYTES
                                                   -----         ------
    Length of font:                                94274        188.548
    Length of font header:                          2208          4.416
    Length of all character headers:               21622         43.244
    Length of total image information:             70444        140.888


              DIGITIZATIONS                             NUMBER
              -------------                             ------
              Start points:                               1349
              Control points:                            16910
              Anchor points:                             16251
                                                        ======
              Total:                                     34510
              -----------------------------------------------
              Characters:                                  712
              Records (a 2048 words):                       47
              -----------------------------------------------
              Raster (grid):                       1000 X  1000
              -----------------------------------------------
              Digitizations per character:                  48
              Image information per character:             198 bytes


--------------------------------------------------------------------------------
```

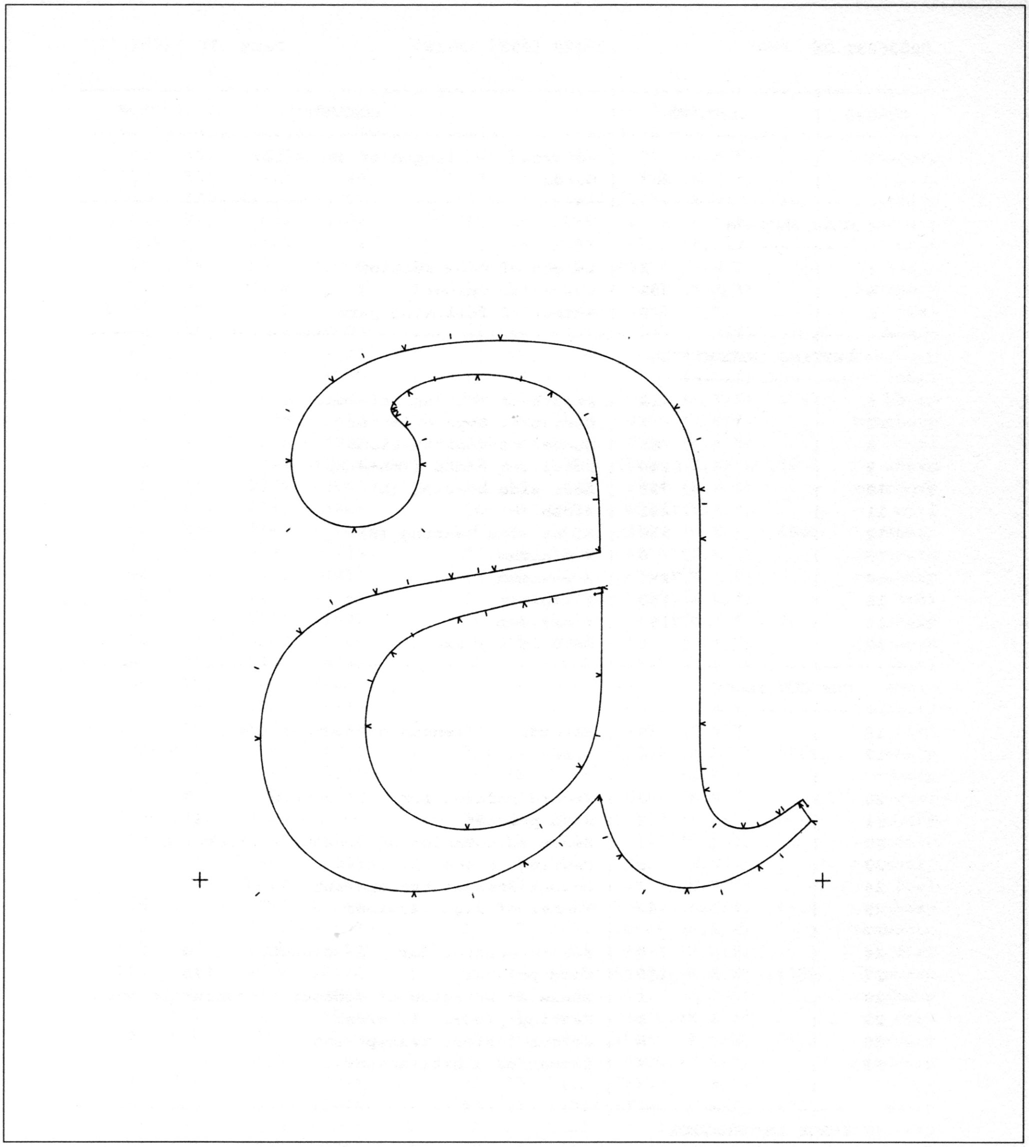


Fig. 186
A CN format character, with marks to show point types

```
U003003T.CN                    20-JAN-1992  11:36           PAGE  15

 ---------|------------------|---------------------------------------------
  WORDNO. |      CONTENT     |                   COMMENT
 ---------|------------------|---------------------------------------------
     1    |                 0 | Records      length of character
     2    |               203 | Words
 ---------|------------------|---------------------------------------------
       NAME SECTION
       ------------
     3    |                 3 | Length of name section
     4    |               301 | Character number
     5    |                 0 | Number of following part
 ---------|------------------|---------------------------------------------
       SETTING INFORMATION
       -------------------
     6    |                12 | Length of setting information
     7    |                 1 | Character type = letter
     8    |                85 | Number of digitizations
     9    |              8240 | Total set width (T=L+W+R)
    10    |               799 | Left side bearing (L)
    11    |              7291 | Width (W)
    12    |               150 | Right side bearing (R)
    13    |                 0 | X-minimum
    14    |              7291 | X-maximum
    15    |              -150 | Y-minimum
    16    |              7153 | Y-maximum
    17    |                 1 | Unit is 1 pixel
 ---------|------------------|---------------------------------------------
       CONTOUR INDEX
       -------------
    18    |                 0 | Records     length of contour index
    19    |                14 | Words
          |                   |
    20    |                 0 | Record pointer for   1. contour:     0
    21    |                 3 | Word pointer                    :    34
    22    |                -1 | Sense of rotation of contour = clockwise
    23    |                 0 | Nesting: outer  1. order
    24    |                 0 | Colour inside: transparent
    25    |                68 | Number of digitizations
          |                   |
    26    |                 0 | Record pointer for   2. contour:     0
    27    |               139 | Word pointer                    :   170
    28    |                 1 | Sense of rotation of contour = counterclockwise
    29    |                 1 | Nesting: inner  1. order
    30    |                 0 | Colour inside: transparent
    31    |                17 | Number of digitizations
          |                   |
 ---------|------------------|---------------------------------------------
       IMAGE INFORMATION
       -----------------
    32    |                 0 | Records     length of image information
    33    |               172 | Words
          |                   |
  34-  35|  -7120   1184     |     1: 12  7119   1033 (K,X,Y)          >startp.
  36-  37|   7292   -941     |     2: 15  7291    790 (K,X,Y)          >anchorp
  38-  39|   6377     49     |     3: 13  6376   -102 (K,X,Y)              ctrl
  40-  41|  -8156 -15000     |     4: 14  8156/ 15000 (K,X/Y) .5437      sharp
  42-  43|   5634    -49     |     5: 15  5633   -102 (K,X,Y)          >anchorp
  44-  45|   5097     49     |     6: 13  5096   -102 (K,X,Y)              ctrl
  46-  47|  -6711 -15000     |     7: 14  6711/ 15000 (K,X/Y) .4474      sharp
 ---------|------------------|---------------------------------------------
```

```
U003003T.CN                  20-JAN-1992  11:36            PAGE  16

  ---------|------------------|-------------------------------------------------
   WORDNO. |     CONTENT      |                    COMMENT
  ---------|------------------|-------------------------------------------------
    48-  49|  4788   -493     |    8: 15  4787    342  (K,X,Y)          >anchorp
    50-  51|  4613    744     |    9: 13  4612    593  (K,X,Y)              ctrl
    52-  53| -8865 -15000     |   10: 14  8865/ 15000  (K,X/Y) .5910      sharp
    54-  55|  4486  -1284     |   11: 15  4485   1133  (K,X,Y)          >anchorp
    56-  57|  3340      1     |   12: 13  3339   -150  (K,X,Y)              ctrl
    58-  59| -8031 -15000     |   13: 14  8031/ 15000  (K,X/Y) .5354      sharp
    60-  61|  2035     -1     |   14: 15  2034   -150  (K,X,Y)          >anchorp
    62-  63|     1      1     |   15: 13     0   -150  (K,X,Y)              ctrl
    64-  65| -6794 -15000     |   16: 14  6794/ 15000  (K,X/Y) .4529      sharp
    66-  67|     1  -2023     |   17: 15     0   1872  (K,X,Y)          >anchorp
    68-  69|     1   2945     |   18: 13     0   2794  (K,X,Y)              ctrl
    70-  71| -7187 -15000     |   19: 14  7187/ 15000  (K,X/Y) .4791      sharp
    72-  73|   543  -3417     |   20: 15   542   3266  (K,X,Y)          >anchorp
    74-  75|  1111   3912     |   21: 13  1110   3761  (K,X,Y)              ctrl
    76-  77| -8794 -15000     |   22: 14  8794/ 15000  (K,X/Y) .5863      sharp
    78-  79|  2524  -4147     |   23: 15  2523   3996  (K,X,Y)          >anchorp
    80-  81|  2885   4207     |   24: 13  2884   4056  (K,X,Y)              ctrl
    82-  83| -7443 -15000     |   25: 14  7443/ 15000  (K,X/Y) .4962      sharp
    84-  85|  3095  -4245     |   26: 15  3094   4094  (K,X,Y)          >anchorp
    86-  87|  4492  -4497     |   27: 15  4491   4346  (K,X,Y)          >anchorp
    88-  89|  4471   5766     |   28: 13  4470   5615  (K,X,Y)              ctrl
    90-  91| -9294 -15000     |   29: 14  9294/ 15000  (K,X/Y) .6196      sharp
    92-  93|  4263  -6188     |   30: 15  4262   6037  (K,X,Y)          >anchorp
    94-  95|  3936   6849     |   31: 13  3935   6698  (K,X,Y)              ctrl
    96-  97| -7247 -15000     |   32: 14  7247/ 15000  (K,X/Y) .4831      sharp
    98-  99|  2860  -6849     |   33: 15  2859   6698  (K,X,Y)          >anchorp
   100- 101|  1722   6849     |   34: 13  1721   6698  (K,X,Y)              ctrl
   102- 103| -5552 -15000     |   35: 14  5552/ 15000  (K,X/Y) .3701      sharp
   104- 105|  1722  -6390     |   36: 15  1721   6239  (K,X,Y)          >anchorp
   106- 107|  1722   6337     |   37: 13  1721   6186  (K,X,Y)              ctrl
   108- 109| -7223 -15000     |   38: 14  7223/ 15000  (K,X/Y) .4815      sharp
   110- 111|  1776  -6284     |   39: 15  1775   6133  (K,X,Y)          >anchorp
   112- 113|  1906  -6157     |   40: 15  1905   6006  (K,X,Y)          >anchorp
   114- 115|  2104   5964     |   41: 13  2103   5813  (K,X,Y)              ctrl
   116- 117| -7132 -15000     |   42: 14  7132/ 15000  (K,X/Y) .4755      sharp
   118- 119|  2104  -5655     |   43: 15  2103   5504  (K,X,Y)          >anchorp
   120- 121|  2104   4828     |   44: 13  2103   4677  (K,X,Y)              ctrl
   122- 123| -6528 -15000     |   45: 14  6528/ 15000  (K,X/Y) .4352      sharp
   124- 125|  1237  -4828     |   46: 15  1236   4677  (K,X,Y)          >anchorp
   126- 127|   403   4828     |   47: 13   402   4677  (K,X,Y)              ctrl
   128- 129| -6618 -15000     |   48: 14  6618/ 15000  (K,X/Y) .4412      sharp
   130- 131|   403  -5701     |   49: 15   402   5550  (K,X,Y)          >anchorp
   132- 133|   403   7304     |   50: 13   402   7153  (K,X,Y)              ctrl
   134- 135| -6889 -15000     |   51: 14  6889/ 15000  (K,X/Y) .4593      sharp
   136- 137|  3169  -7304     |   52: 15  3168   7153  (K,X,Y)          >anchorp
   138- 139|  4844   7304     |   53: 13  4843   7153  (K,X,Y)              ctrl
   140- 141| -7677 -15000     |   54: 14  7677/ 15000  (K,X/Y) .5118      sharp
   142- 143|  5462  -6395     |   55: 15  5461   6244  (K,X,Y)          >anchorp
   144- 145|  5814   5878     |   56: 13  5813   5727  (K,X,Y)              ctrl
   146- 147| -8640 -15000     |   57: 14  8640/ 15000  (K,X/Y) .5760      sharp
   148- 149|  5814  -4677     |   58: 15  5813   4526  (K,X,Y)          >anchorp
   150- 151|  5814  -2222     |   59: 15  5813   2071  (K,X,Y)          >anchorp
   152- 153|  5814   1625     |   60: 13  5813   1474  (K,X,Y)              ctrl
   154- 155| -8495 -15000     |   61: 14  8495/ 15000  (K,X/Y) .5663      sharp
   156- 157|  5870  -1344     |   62: 15  5869   1193  (K,X,Y)          >anchorp
   158- 159|  5970    841     |   63: 13  5969    690  (K,X,Y)              ctrl
  ---------|------------------|-------------------------------------------------
```

```
U003003T.CN                   20-JAN-1992  11:36            PAGE  17

 ---------|------------------|---------------------------------------------------
  WORDNO. |     CONTENT      |                    COMMENT
 ---------|------------------|---------------------------------------------------
  160- 161|  -6962 -15000    |    64: 14  6962/ 15000  (K,X/Y)  .4641        sharp
  162- 163|   6381   -841    |    65: 15  6380    690  (K,X,Y)            >anchorp
  164- 165|   6633    841    |    66: 13  6632    690  (K,X,Y)                ctrl
  166- 167|  -9864 -15000    |    67: 14  9864/ 15000  (K,X/Y)  .6576        sharp
  168- 169|   7120  -1184    |    68: 15  7119   1033  (K,X,Y)            >anchorp
  170- 171|  -4514   4035    |    69: 12  4513   3884  (K,X,Y)            >startp.
  172- 173|   2925   3760    |    70: 13  2924   3609  (K,X,Y)                ctrl
  174- 175|  -7919 -15000    |    71: 14  7919/ 15000  (K,X/Y)  .5279        sharp
  176- 177|   2341  -3549    |    72: 15  2340   3398  (K,X,Y)            >anchorp
  178- 179|   1390   3204    |    73: 13  1389   3053  (K,X,Y)                ctrl
  180- 181|  -7803 -15000    |    74: 14  7803/ 15000  (K,X/Y)  .5202        sharp
  182- 183|   1390  -2184    |    75: 15  1389   2033  (K,X,Y)            >anchorp
  184- 185|   1390    822    |    76: 13  1389    671  (K,X,Y)                ctrl
  186- 187|  -6629 -15000    |    77: 14  6629/ 15000  (K,X/Y)  .4419        sharp
  188- 189|   2738   -822    |    78: 15  2737    671  (K,X,Y)            >anchorp
  190- 191|   3738    822    |    79: 13  3737    671  (K,X,Y)                ctrl
  192- 193|  -6873 -15000    |    80: 14  6873/ 15000  (K,X/Y)  .4582        sharp
  194- 195|   4255  -1641    |    81: 15  4254   1490  (K,X,Y)            >anchorp
  196- 197|   4514   2052    |    82: 13  4513   1901  (K,X,Y)                ctrl
  198- 199|  -7100 -15000    |    83: 14  7100/ 15000  (K,X/Y)  .4733        sharp
  200- 201|   4514  -2866    |    84: 15  4513   2715  (K,X,Y)            >anchorp
  202- 203|   4514  -4035    |    85: 15  4513   3884  (K,X,Y)            >anchorp
 -------------------------------------------------------------------------------

                              GRAND TOTAL
                              -----------
                                                   WORDS        KBYTES
                                                   -----        ------
    Length of font:                                94276       188.552
    Length of font header:                          2208         4.416
    Length of all character headers:               21622        43.244
    Length of total image information:             70446       140.892

              DIGITIZATIONS                            NUMBER
              -------------                            ------
              Start points:                              1349
              Control points:                            8455
              Sharpnesses:                               8455
              Anchor points:                            16252
                                                       ======
              Total:                                    34511
              ---------------------------------------------
              Characters:                                 712
              Records (a 2048 words):                      47
              ---------------------------------------------
              Raster (grid):                    15000 X 15000
              ---------------------------------------------
              Digitizations per character:                 48
              Image information per character:            198 bytes

 -------------------------------------------------------------------------------
```

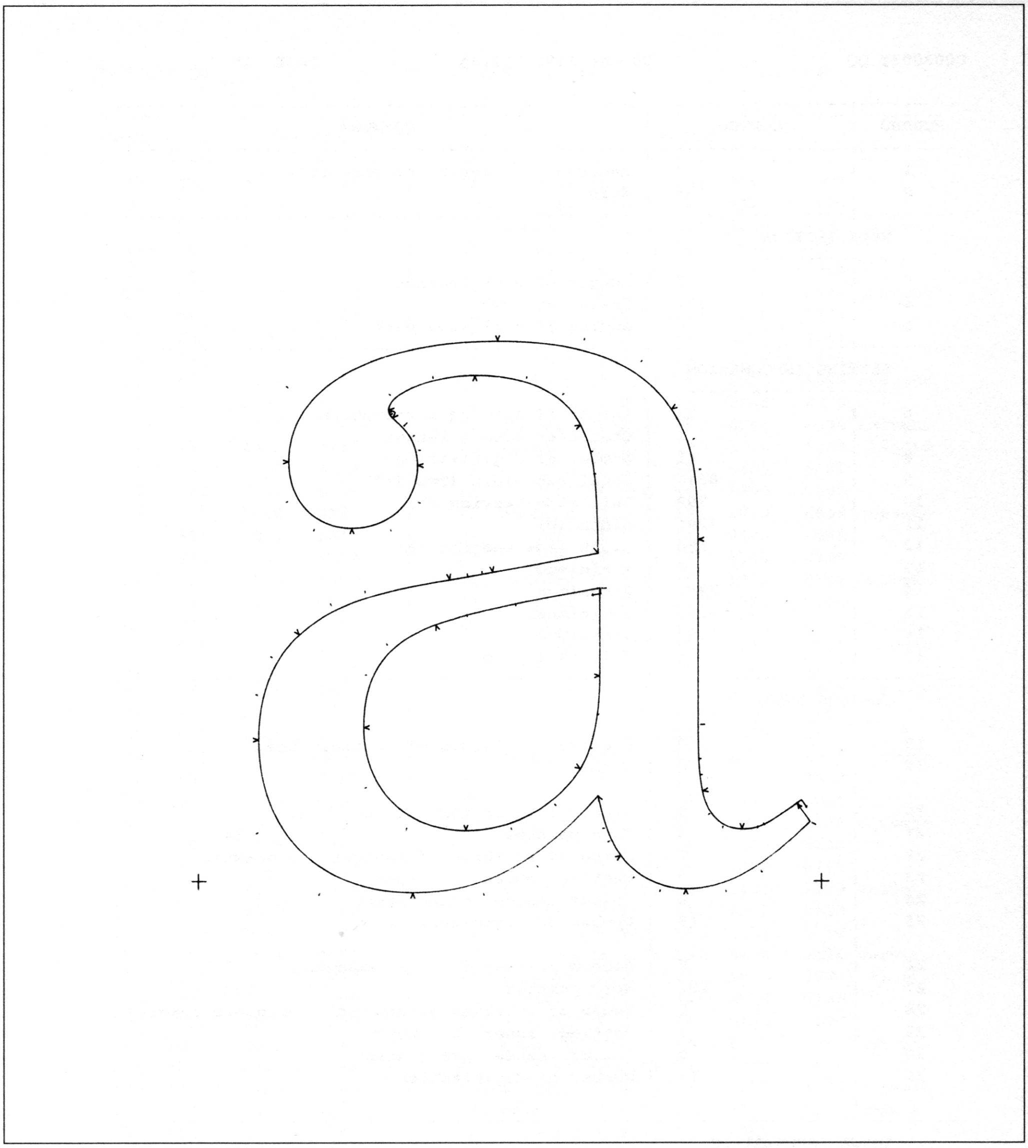
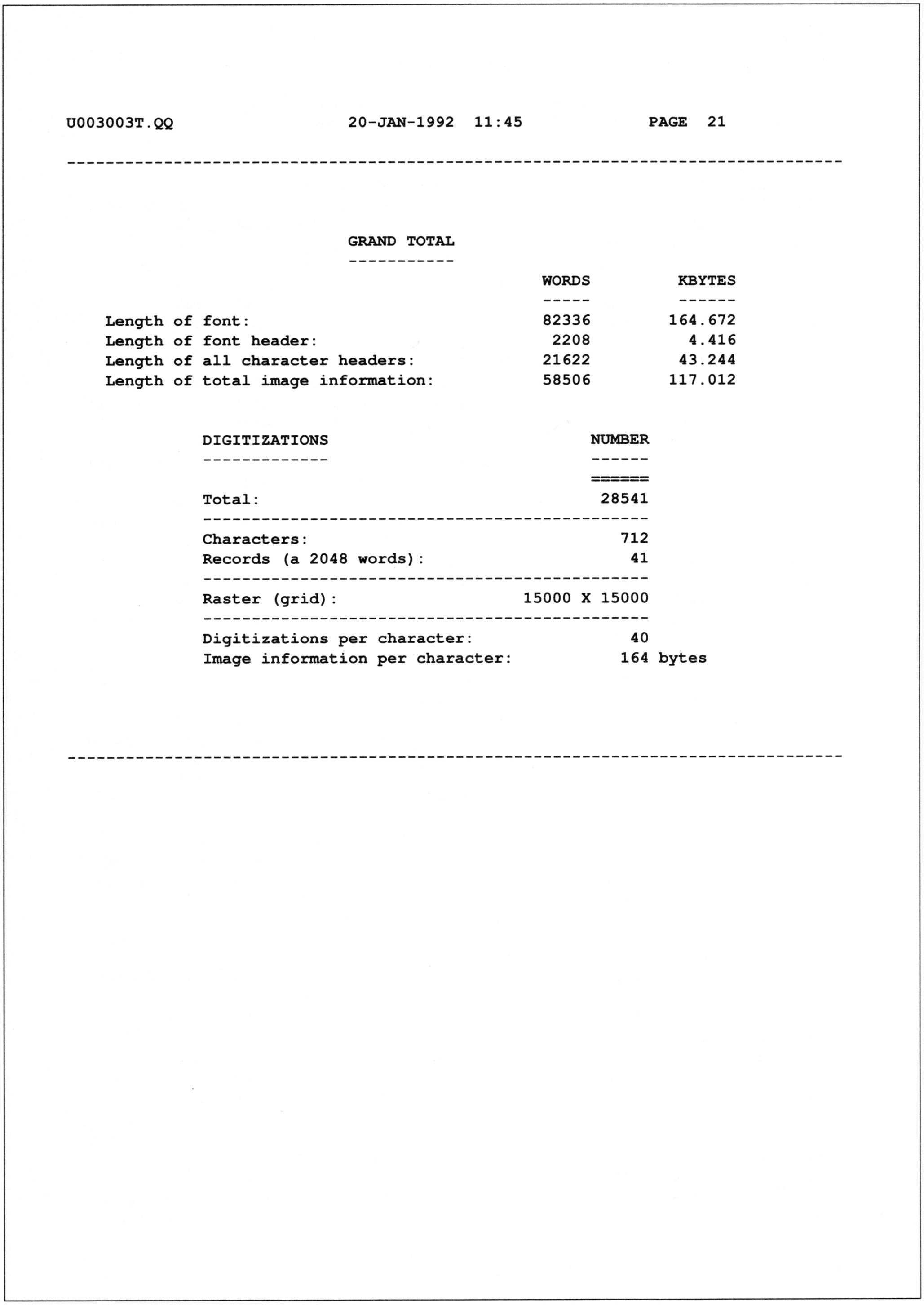


Fig. 187
A QQ format character, with marks to show point types

```
U003003T.QQ                    20-JAN-1992  11:45            PAGE  15

 ---------|------------------|-------------------------------------------------
  WORDNO. |      CONTENT     |                       COMMENT
 ---------|------------------|-------------------------------------------------
     1    |                 0 | Records      length of character
     2    |               155 | Words
 ---------|------------------|-------------------------------------------------
       NAME SECTION
       ------------
     3    |                 3 | Length of name section
     4    |               301 | Character number
     5    |                 0 | Number of following part
 ---------|------------------|-------------------------------------------------
       SETTING INFORMATION
       -------------------
     6    |                12 | Length of setting information
     7    |                 1 | Character type = letter
     8    |                61 | Number of digitizations
     9    |              8240 | Total set width (T=L+W+R)
    10    |               799 | Left side bearing (L)
    11    |              7291 | Width (W)
    12    |               150 | Right side bearing (R)
    13    |                 0 | X-minimum
    14    |              7291 | X-maximum
    15    |              -150 | Y-minimum
    16    |              7153 | Y-maximum
    17    |                 1 | Unit is 1 pixel
 ---------|------------------|-------------------------------------------------
       CONTOUR INDEX
       -------------
    18    |                 0 | Records     length of contour index
    19    |                14 | Words
          |                   |
    20    |                 0 | Record pointer for   1. contour:     0
    21    |                 3 | Word pointer                    :    34
    22    |                -1 | Sense of rotation of contour = clockwise
    23    |                 0 | Nesting: outer  1. order
    24    |                 0 | Colour inside: transparent
    25    |                49 | Number of digitizations
          |                   |
    26    |                 0 | Record pointer for   2. contour:     0
    27    |               101 | Word pointer                    :   132
    28    |                 1 | Sense of rotation of contour = counterclockwise
    29    |                 1 | Nesting: inner  1. order
    30    |                 0 | Colour inside: transparent
    31    |                12 | Number of digitizations
          |                   |
 ---------|------------------|-------------------------------------------------
       IMAGE INFORMATION
       -----------------
    32    |                 0 | Records     length of image information
    33    |               124 | Words
          |                   |
   34-  35|  -7120   1184    |      1. startp. (K:X,Y) 12:  7119   1033  moveto
   36-  37|  -7292   -941    |      2. anchorp (K:X,Y) 13:  7291    790  lineto
   38-  39|   6377     49    |      3.    ctrl (K:X,Y) 14:  6376   -102
   40-  41|   5634    -49    |      4. anchorp (K:X,Y) 15:  5633   -102 curveto
   42-  43|   5323     49    |      5.    ctrl (K:X,Y) 14:  5322   -102
   44-  45|   4676    479    |      6.    ctrl (K:X,Y) 14:  4675    328
   46-  47|   4486  -1284    |      7. anchorp (K:X,Y) 15:  4485   1133 curveto
 ---------|------------------|-------------------------------------------------
```

```
U003003T.QQ                 20-JAN-1992  11:45             PAGE  16

 ---------|------------------|-----------------------------------------------
  WORDNO. |     CONTENT      |                   COMMENT
 ---------|------------------|-----------------------------------------------
   48-  49|   3798      514  |    8.    ctrl  (K:X,Y) 14:  3797    363
   50-  51|   2628        1  |    9.    ctrl  (K:X,Y) 14:  2627   -150
   52-  53|   2035       -1  |   10. anchorp  (K:X,Y) 15:  2034   -150 curveto
   54-  55|   1131        1  |   11.    ctrl  (K:X,Y) 14:  1130   -150
   56-  57|      1     1110  |   12.    ctrl  (K:X,Y) 14:     0    959
   58-  59|      1    -2023  |   13. anchorp  (K:X,Y) 15:     0   1872 curveto
   60-  61|      1     2545  |   14.    ctrl  (K:X,Y) 14:     0   2394
   62-  63|    391     3362  |   15.    ctrl  (K:X,Y) 14:   390   3211
   64-  65|   1343     3950  |   16.    ctrl  (K:X,Y) 14:  1342   3799
   66-  67|   2524    -4147  |   17. anchorp  (K:X,Y) 15:  2523   3996 curveto
   68-  69|   2885     4207  |   18.    ctrl  (K:X,Y) 14:  2884   4056
   70-  71|   3095    -4245  |   19. anchorp  (K:X,Y) 15:  3094   4094 curveto
   72-  73|  -4492    -4497  |   20. anchorp  (K:X,Y) 13:  4491   4346  lineto
   74-  75|   4492     6053  |   21.    ctrl  (K:X,Y) 14:  4491   5902
   76-  77|   3691     6849  |   22.    ctrl  (K:X,Y) 14:  3690   6698
   78-  79|   2860    -6849  |   23. anchorp  (K:X,Y) 15:  2859   6698 curveto
   80-  81|   2304     6849  |   24.    ctrl  (K:X,Y) 14:  2303   6698
   82-  83|   1722     6500  |   25.    ctrl  (K:X,Y) 14:  1721   6349
   84-  85|   1722    -6390  |   26. anchorp  (K:X,Y) 15:  1721   6239 curveto
   86-  87|   1722     6337  |   27.    ctrl  (K:X,Y) 14:  1721   6186
   88-  89|   1776    -6284  |   28. anchorp  (K:X,Y) 15:  1775   6133 curveto
   90-  91|  -1906    -6157  |   29. anchorp  (K:X,Y) 13:  1905   6006  lineto
   92-  93|   2104     5964  |   30.    ctrl  (K:X,Y) 14:  2103   5813
   94-  95|   2104    -5655  |   31. anchorp  (K:X,Y) 15:  2103   5504 curveto
   96-  97|   2104     5334  |   32.    ctrl  (K:X,Y) 14:  2103   5183
   98-  99|   1638     4828  |   33.    ctrl  (K:X,Y) 14:  1637   4677
  100- 101|   1237    -4828  |   34. anchorp  (K:X,Y) 15:  1236   4677 curveto
  102- 103|    844     4828  |   35.    ctrl  (K:X,Y) 14:   843   4677
  104- 105|    403     5360  |   36.    ctrl  (K:X,Y) 14:   402   5209
  106- 107|    403    -5701  |   37. anchorp  (K:X,Y) 15:   402   5550 curveto
  108- 109|    403     6396  |   38.    ctrl  (K:X,Y) 14:   402   6245
  110- 111|   1829     7304  |   39.    ctrl  (K:X,Y) 14:  1828   7153
  112- 113|   3169    -7304  |   40. anchorp  (K:X,Y) 15:  3168   7153 curveto
  114- 115|   4373     7304  |   41.    ctrl  (K:X,Y) 14:  4372   7153
  116- 117|   5814     6320  |   42.    ctrl  (K:X,Y) 14:  5813   6169
  118- 119|   5814    -4677  |   43. anchorp  (K:X,Y) 15:  5813   4526 curveto
  120- 121|  -5814    -2222  |   44. anchorp  (K:X,Y) 13:  5813   2071  lineto
  122- 123|   5814     1296  |   45.    ctrl  (K:X,Y) 14:  5813   1145
  124- 125|   6120      841  |   46.    ctrl  (K:X,Y) 14:  6119    690
  126- 127|   6381     -841  |   47. anchorp  (K:X,Y) 15:  6380    690 curveto
  128- 129|   6633      841  |   48.    ctrl  (K:X,Y) 14:  6632    690
  130- 131|   7120    -1184  |   49. anchorp  (K:X,Y) 15:  7119   1033 curveto
  132- 133|  -4514     4035  |   50. startp.  (K:X,Y) 12:  4513   3884  moveto
  134- 135|   2965     3767  |   51.    ctrl  (K:X,Y) 14:  2964   3616
  136- 137|   1824     3368  |   52.    ctrl  (K:X,Y) 14:  1823   3217
  138- 139|   1390     2677  |   53.    ctrl  (K:X,Y) 14:  1389   2526
  140- 141|   1390    -2184  |   54. anchorp  (K:X,Y) 15:  1389   2033 curveto
  142- 143|   1390     1593  |   55.    ctrl  (K:X,Y) 14:  1389   1442
  144- 145|   2145      822  |   56.    ctrl  (K:X,Y) 14:  2144    671
  146- 147|   2738     -822  |   57. anchorp  (K:X,Y) 15:  2737    671 curveto
  148- 149|   3397      822  |   58.    ctrl  (K:X,Y) 14:  3396    671
  150- 151|   4514     1750  |   59.    ctrl  (K:X,Y) 14:  4513   1599
  152- 153|   4514    -2866  |   60. anchorp  (K:X,Y) 15:  4513   2715 curveto
  154- 155|  -4514    -4035  |   61. anchorp  (K:X,Y) 13:  4513   3884  lineto
          |                  |
 ---------|------------------|-----------------------------------------------
```



```
U003003T.QQ              20-JAN-1992  11:45              PAGE  21
--------------------------------------------------------------------------------


                            GRAND TOTAL
                            -----------
                                                  WORDS         KBYTES
                                                  -----         ------
    Length of font:                               82336        164.672
    Length of font header:                         2208          4.416
    Length of all character headers:              21622         43.244
    Length of total image information:            58506        117.012


              DIGITIZATIONS                           NUMBER
              -------------                           ------
                                                      ======
              Total:                                   28541
              ----------------------------------------------
              Characters:                                712
              Records (a 2048 words):                     41
              ----------------------------------------------
              Raster (grid):                   15000 X 15000
              ----------------------------------------------
              Digitizations per character:                40
              Image information per character:           164 bytes



--------------------------------------------------------------------------------
```

Description of the II format

by Jochen Lau

Introduction

This appendix describes URW's proprietary II format, the II being an abbreviation for «Intelligent IKARUS format» or «Instructed IKARUS format» alternatively: «IKARUS format with Instructions». II is the source format for production of intelligent machine formats such as PostScript Type1, TrueType, etc.

II *stands for* I*nstructed* I*karus*

File Structure

Each II font file is a binary file. Files have no blocking or records, and a random access method is used.

Binary file divided into named sections

The actual font information is stored in various named sections, not all of which need be present in every font. A benefit of this approach is that it allows users to store just IK data without instructions in an II font. In addition, users can add further sections without having to modify the programs working with the II format: these programs can continue to work with sections they understand.

The sections of an II font are accessed via the File Address Block (FAB). The FAB forms part of the Control and Address Section (CASE), which also comprises a 12-byte fixed data record at the start of the file, containing computer-specific control information.

The contents of the sections is described in more detail in this appendix.

The following data types are used in an II font:

ASCII	Unsigned 8-bit value for representing text
ubyte	Unsigned 8-bit integer value
short	Signed 16-bit integer value
ushort	Unsigned 16-bit integer value
long	Signed 32-bit integer value
ulong	Unsigned 32-bit integer value

II fonts observe the following conventions:

- All length values are given in bytes.
- Counting always starts at 0 (zero).
- All sections commence at a four-byte word boundary.
- All pointers are of the ulong type and give a byte address.
- Where not stated otherwise, all pointers refer to the start of sections.

Control and Address Section

The **Control and Address Section** consists of 12 bytes of computer-specific control information, a 4-byte length field for the FAB, plus the File Address Block. The FAB contains the names, addresses and lengths of all the sections in the font. The sections do not have to appear in the file in any predefined order, nor must entries in the FAB be sorted. The CASE section is always located at the start of the file.

Byte	Type	Meaning
0,1 2,3	ASCII ushort	Name of the format, here II. Byte sequence. «0x0000», for example, indicates that the font was generated in a byte sequence as used on the well-known family of Motorola 68xxx processors. «0x0101» means that the byte sequence corresponds to the Vax or Intel 80x86 processors.
4..7	long	Checksum giving the sum of all subsequent bytes in the font, interpreted as signed quantities. All programs which change the font should update the Checksum prior to closing the font file. When the font is copied to a different computer system, the Checksum ensures that the operation is executed correctly.
8..11	ulong	Length of the font file
12..15	ulong	Length of the FAB (multiple of 12, as the entry for a section is 12 bytes long)
		File Address Block
16..19	ASCII	Name of the first section
20..23	ulong	Address of the first section
24..27	ulong	Length of the first section (actual length, i.e. rounded to a multiple of 4 bytes)
28..31	ASCII	Name of the second section
32..35	ulong	Address of the second section
36..39	ulong	Length of the second section
. . .	. . .	. . .

The following names are presently reserved for sections:

- FDBL Format Definition Block
- FINF Font Information
- FOHD Font Header

- FORT Font Reference Table
- FOGL Font Guide Lines
- IKDR IK Directory (directory for IK data)
- IKDT IK Data (IK character data)
- SCAL Intelligent Font Scaling Data (NIMBUS)

Format Definition Block (FDBL)

The **Format Definition Block** contains a number of values either not included in the general specifications at the start of the font or differ from these values.

Byte	Type	Meaning
0, 1	ushort	Version number of the format. The first byte gives the figure before the decimal point, the second the figure after the point. Version 1.1, for instance, would be coded with a 1 in both bytes.
2, 3	ushort	Number of bytes in the character names; 2, 4 or 6 bytes.
4, 5	ushort	Number of bytes used to specify indices in instructions. If all referenced point numbers and indices in the FORT are smaller than 256, they can be coded with one byte.
6, 7	ushort	Length of the character header in bytes (28 in the current version).

Font Information (FINF)

This section contains general information about the typeface, which is not available from IK format but needed for further conversions as to Type1 or TrueType.

The **Font Information** data consist of so-called «**information blocks**».

An information block is composed of three parts:

- the code number which specifies the kind of information
- the length of the information block in bytes
- the information itself.

This structure enables a flexible handling of the informations within the programs. The needed or not needed informations are identified by their code; by means of the length information it is possible to read over unknown blocks. This guarantees the possibility to enlarge the information set later on, so that also old program versions may work with the extended information set without changing the software.

Information block

Byte	Type	Meaning
0,1	ushort	information code
2,3	ushort	length of the information block in bytes, rounded off to a multiple of four bytes
4, ..., n	long, ASCII	information

The information may be a number or an ASCII string, so that the programs must take care how to interprete the input.

If the information consists of ASCII characters, the string may be filled with character 0 to match a four byte boundary.

In the following the informations supported by URW will be specified and their structure will be described.

Information codes 16 to 31 deal with global data for the font, codes 32 to 47 are reserved for additional information needed to convert to Type1, codes 64 to 79 to convert the FONDs by IKARUS M, codes 128 to 145 to convert to PFM and codes 256 to 270 to convert to TrueType.

Code	Length	Information	Type
16	8 Byte	Ascender	long
17	4 + n Byte	Comment	ASCII
18	4 + n Byte	Copyright	ASCII
19	4 + n byte	FamilyName	ASCII
20	4 + n Byte	FontName	ASCII

Code	Length	Information	Type
21	4 + n Byte	FullName	ASCII
22	4 + n Byte	IsFixedPitch	ASCII
23	4 + n Byte	Notice	ASCII
24	8 Byte	UnderlinePosition	long
25	8 byte	UnderlineThickness	long
26	4 + n Byte	Version	ASCII
27	4 + n Byte	Weight	ASCII
28	8 Byte	AccentOffset	long
29	8 byte	Descender	long
30	8 Byte	CapHeight	long
31	8 Byte	XHeight	long
32	4 + n Byte	Identifier	ASCII
33	8 Byte	Uinque ID	long
34	8 byte	FigureSize	long
64	8 Byte	FondID	long
65	4 + n Byte	FondName	ASCII
66	4 + n byte	MacFileName	ASCII
67	8 Byte	MacStyle	long
68	8 Byte	FondAscender	long
69	8 Byte	FondLeading	long
70	8 Byte	FondDescender	long
71	8 Byte	Bodysize	long
128	4 + n Byte	FaceName	ASCII
129	8 Byte	PCWeight	long
130	4 + n Byte	PitchAndFamily	ASCII
131	8 Byte	SuperScript	long
132	8 Byte	SubScript	long
133	8 Byte	SuperScriptSize	long
134	8 Byte	SubScriptSize	long
135	8 Byte	UnderlineOffset	long
136	8 Byte	UnderlineWidth	long
137	8 Byte	DoubleUpperUnderlineOffset	long
138	8 Byte	DoubleLowerUnderlineOffset	long
139	8 Byte	DoubleUpperUnderlineWidth	long
140	8 Byte	DoubleLowerUnderlineWidth	long
141	8 byte	StrikeOutOffset	long
142	8 Byte	StrikeOutWidth	long
143	8 Byte	InternalLeading	long
144	8 Byte	ExternalLeading	long
145	8 Byte	Slant	long
256	8 byte	WidthClass	long
257	8 Byte	FamilyClass	long
258	4 + n Byte	Panose	ASCII
259	4 + n Byte	VendID	ASCII
260	8 Byte	TypoLineGap	long
261	8 Byte	TypoAscender	long
262	8 Byte	TypoDescender	long
263	8 Byte	SubscriptXSize	long
264	8 Byte	SuperscriptXSize	long
265	8 Byte	AscenderHHEA	long
266	8 Byte	DescenderHHEA	long
267	8 Byte	LineGapHHEA	long
268	4 + n Byte	TrueTypeID	ASCII
269	4 + n Byte	FontFamilyName	ASCII
270	4 + n Byte	SubfamilyName	ASCII

Font Header (FOHD)

The **Font Header** contains general font information. All entries must be made by the program generating the font. Where parameters do not have to be defined explicitly by users, the programs enter suitable default values in the font header.

Byte	Type	Meaning
0, ..., 79	ASCII	Name of the font
80, ..., 85	ushort	Production date (binary format, day, month, year)
86, ..., 91	ushort	Date of last change (format as above)
92, ..., 95	long	em square in smallest units (1/100 mm) When the font is changed by a software program, the software should enter the body size here automatically if, say, it is operating with a standardized cap height but a variable body size. User can then rely on the font header entries being correct when the font is next opened.
96, ..., 99	long	Cap height in smallest units (1/100 mm)
100, ..., 103	long	x-height in smallest units (1/100 mm)
104, ..., 107	long	Distance from base line to lower body line in smallest units (1/100 mm)
108, 109	short	Italic angle of caps (1/10 degrees)
110, 111	short	Italic angle of lower case characters (1/10 degrees)
112, ..., 115	long	X-minimum of font bounding box
116, ..., 119	long	X-maximum of font bounding box
120, ..., 123	long	Y-minimum of font bounding box
124, ..., 127	long	Y-maximum of font bounding box When the font is changed by a software program, the software must also update the bounding box automatically.
128, ..., 131	long	Number of characters in this font

Other font-specific information required by various formats (such as Family Name, Weight, etc. in Type 1) is described in another section.

If for any reason certain data is not available in the font, the entries are coded with -1, e.g. x-height for logo/signet fonts or non-Latin alphabets.

Font Reference Table (FORT)

Reference values which are evaluated during rasterizing.

The **Font Reference Table** contains control data required to interpret instructions. The table defines reference values for stem widths, ascenders/descenders, plus permitted tolerance ranges. All values are accessed via the index in this table. Entries are numbered consecutively, starting at 0. Each entry consists of a reference value and a tolerance value. When a font is rasterized, the reference value and the value computed from the font data are both rounded to the target resolution. If the computed value differs from the reference value by less than a defined tolerance value, the reference value overwrites the data-computed value.

Instructions determine exactly how the reference values are applied in each case.

Byte	Type	Meaning
0, 1	short	First reference value, index 0, given in units of the body size
2, 3	short	Tolerance, given in 1/128 units of the target grid
4, 5	short	Second reference value, index 1
6, 7	short	Tolerance for second value
.	.	.
.	.	.
.	.	.

Font Guide Lines (FOGL)

Font lines such as cap height

The **Font Guide Line** Table contains the numerical values of font guide lines, control data required to interpret instructions. All values are accessed via the index in this table. Entries are numbered consecutively, starting at 0, with each entry consisting of a reference and tolerance value. The instructions determine exactly how the reference values are applied in each case.

Byte	Type	Meaning
0, 1	short	First font line (cap height) The value is given in the units of the body size.
2, 3	short	Tolerance for the first font line
4, 5	short	Second font line (x-height), index 1
6, 7	short	Tolerance for second value
8,9	short	Third font line (baseline), index 2
10,11	short	Tolerance for third value
12,13	short	Fourth font line (descender line), index 3
14,15	short	Tolerance for fourth value
.	.	.
.	.	.
.	.	.

Though some guidelines have already been defined in the font header, they reappear in this table ensuring that identical access methods are used for all indexed references.

IKARUS Directory (IKDR)

This section contains addresses of IK format characters in a font. (n + 8) bytes are provided for each character, n being the length of the character number defined in the FDBL (2, 4 or 6 bytes).

Byte	Type	Meaning
0, ..., n-1		Character number corresponding to the length n given in the FDBL.
n + 0, ..., 3	ulong	Address of the first character relative to the start of the IKDT
n + 4, ..., 7	ulong	Length of the IK data of the first character
.	.	.
.	.	.
.	.	.

IK Character Data (IKDT)

This section contains the IK character data proper. Character structure is virtually the same as in IKARUS Version 4, the main difference being that length data has been adapted to the general structure of an II font. Each character has a header, a contour index and image information plus instructions. These follow as a block immediately after the image information.

Byte	Type	Meaning
		Character Header
0, ..., n-1		Character number. Required to enable deleted character information to be accessed, as this can no longer be referenced from the character directory.
n + 0, ..., 3	ulong	Length of the character in bytes, used as above.
n + 4, 5	ushort	Control word
n + 6, .., 9	ulong	Number of digitizations
n + 10,11	short	Total width
n + 12,13	short	Left side bearing
n + 14, 15	short	X-minimum of digitizations. X = Xmin is the right edge of the left side bearing. As a rule, this value is 0, but new programs should not necessarily assume that X-minimum is always 0.
n + 16, 17	short	X-maximum of digitizations
n + 18, 19	short	Y-minimum of digitizations
n + 20, 21	short	Y-maximum of digitizations
n + 22, 23	short	Unit. If unit = 1, the previous six values and all digitizations of the character are specified in 1/100 mm units or in pixels. Unit = 2 means that these values must be multiplied by 2 before they are used. This is used to enable storage of digitizations in 16-bit integer format.

Byte	Type	Meaning
		Contour Index
n + 24, ..., 27	ulong	Length of contour index in bytes (k)
n + 28, ..., 31	ulong	Index of start point of contour. The index (number of the digitization) begins at 0.
n + 32, 33	short	Sense of rotation of first contour: -1 clockwise, 0 open contour 1 anti-clockwise.
n + 34, 35	short	Nesting hierarchy
n + 36, 37	ushort	Color
n + 38, 39	ushort	Number of digitizations in first contour
....		Further contours
		Image Information
n + k + 0,...,3	long	Length of image information (number of digitizations *4 + 4) in bytes (b)
n + k + 4,...,7	short	x/y-coordinate of first digitization; point type coded in the two sign characters as customary in IKARUS V4.
....		Further digitizations for the entire character.
		Instructions
n+k+b+0,...,3	long	Length of the instructions in bytes
n+k+b+4,...,7	short	Instructions (data)
. . .	. . .	. . .

Character Instruction Data

II is a base format from which other machine formats are derived.

The II format is not primarily intended for processing by a RIP, but is more a basis data format, a source format used for automatic generation of such common formats as PostScript Type1 or TrueType. We will continue here with the general approach of describing character *parts* requiring extra control. Machine formats frequently have integrated statements to handle these parts. Intellifont, for instance, uses «Associated point», and TrueType the statements «MDRP» or «MIRP» to fit relative stem edges, whether straight stems or character widths, to the target grid.

The II format uses so-called instruction blocks to describe character sections. An instruction block consists of these three parts:

- a code number indicating which instruction this is,
- a length field defining the length of the instruction block,
- the instruction proper.

Extendable instruction-Set

This structure gives programs a high degree of flexibility in handling instructions. The code allows programs to identify known or unknown instructions, and the length field can instruct the program to ignore those unknown. The instruction set can thus be easily extended at a later date, and older program versions will work with a new extended instruction set without prior alteration.

Following the instruction block's length field are the instructions themselves, grouped by type and character. For example, the code for horizontal curve extrema would be followed by the instruction block's length field, followed, in turn, by the indices of all horizontal extrema in the character.

References to IKDT, FORT and FOGL

Depending on their context, indices used in the instructions will reference either character image information, the Font Reference Table (FORT) or the Font Guide Lines (FOGL). Each instruction's structure determines where the indices point; information which must be known to the application program. All indices begin at 0.

The length of the instruction block is the number of instructions available with a specified code (e.g., number of horizontal stems) for this character multiplied by the number of bytes required for the instruction, plus 4 bytes (first two words in the instruction block).

Byte	Type	Meaning
		Instruction-Code
0,1	ushort	Instruction-Code
2,3	ushort	Length of the instruction block in bytes
4, ..., n	ushort	Instructions

Below is a list of instructions currently supported by URW, with an overview of their structure.

All instruction block entries are of the ushort type. Indices for IKARUS knots point to image information, indices for the Font Reference Table (FORT) or Font Guide Lines (FOGL) point to these tables. The entry 0xFFFF indicates no index is available. Length is given in bytes.

Instructions are interpreted differently depending on which application program is used. NIMBUS evaluates some instructions differently than a program converting II format to Type1, which must re-interpret instructions to generate PostScript hints.

Code	Length	Meaning	Structure of Instructions
1	2 Byte	Absolute cross stem edge	Index of point
2	2 Byte	Absolute stem edge	Index of point
3	4 Byte	Relative cross stem edge (Relative to a font line)	Index of point Index in FOGL
4	6 Byte	Relative cross stem edge	Index of point Index of reference point Index in FORT (reference stem width)
5	6 Byte	Relative stem edge	Index of point Index of refernce point Index in FORT refernce point

Code	Length	Meaning	Structure of Instructions
6	2 Byte	Horizontal curve extremum	Index of extreme point
7	2 Byte	Vertical curve extremum	Index of extreme point
8	4 Byte	Relative bow edge (relative to a font line)	Index of point Index in FOGL
9	6 Byte	Relative bow edge	Index of point Index of reference point Index in FORT (reference stem width)
10	6 Byte	Relative rounding edge	Index of point Index of reference point Index in FORT (reference stem width)
11	6 Byte	Relative horizontal point	Index of point Index of reference point Index in FORT (reference stem width)
12	6 Byte	Relative vertical point	Index of point Index of reference point Index in FORT (reference stem width)
13	2 Byte	Horizontal slant	Index of peak
14	4 Byte	Vertical slant	Index of peak Index in FOGL
15	8 Byte	Serif	Index y-anchor Index x-anchor (stem attachment) Index in FORT (reference length) Index in FOGL (baseline) for the y-anchor
16	6 Byte	Cross serif	Index start of serif Index end of serif Index in FORT (reference length)

Code	Length	Meaning	Structure of Instructions
17	6 Byte	Shallow curves	Start index End index Index in FORT (grid below which a straight line is produced)).
18	6 Byte	Horizontal counter	Index of point Index of reference point Index in FORT (reference width)
19	6 Byte	Vertical counter	Index of point Index of reference point Index in FORT (reference width)
20	2 Byte	Spot (left)	Index of point to be corrected: left or below is black.
21	2 Byte	Spot (right)	Index of point to be corrected: right or above is black.
22	8 Byte	Angled stem	Start index of relative edge End index of relative edge Start index of absolute edge End index of absolute edge

Data to Support Intelligent Font Scaling (SCAL)

Data in this section is evaluated by rasterizers such as NIMBUS, but can also be transferred into other formats (Type1) as part of format conversions.

Byte	Type	Meaning
0, 3	ulong	Identification number of the font in caching, corresponds to the Unique ID in PostScript.
4, 5	ushort	X-resolution of the grid to which the font was fitted. If the body size given in the font header is 15,000, for example, and the entry here is 1,000, the x-coordinates of some of the points (e.g. curve extrema) will be multiples of 15,000/1,000 = 1/5.
6, 7	ushort	Y-resolution of the grid to which the font was fitted. If the body size given in the font header is 15,000, for example, and the entry here is 1,000, the y-coordinates of some of the points (e.g. curve extrema) will be multiples of 15,000/1,000 = 1/5.
8,9	ushort	Canonical stem width in units of the body size. This entry is always evaluated if the grid is smaller than the value specified in SCAL. This causes the stems of upper and lower case characters to be adjusted to a uniform width.
10,11	ushort	Tolerance value for canonical stem width in 1/128 of the target grid.
12,13	ushort	Number of pixels in the target grid em square below which the canonical stem width is to be applied.
14,15	ushort	Number of pixels in the target grid em square below which the program will operate with dropout control.

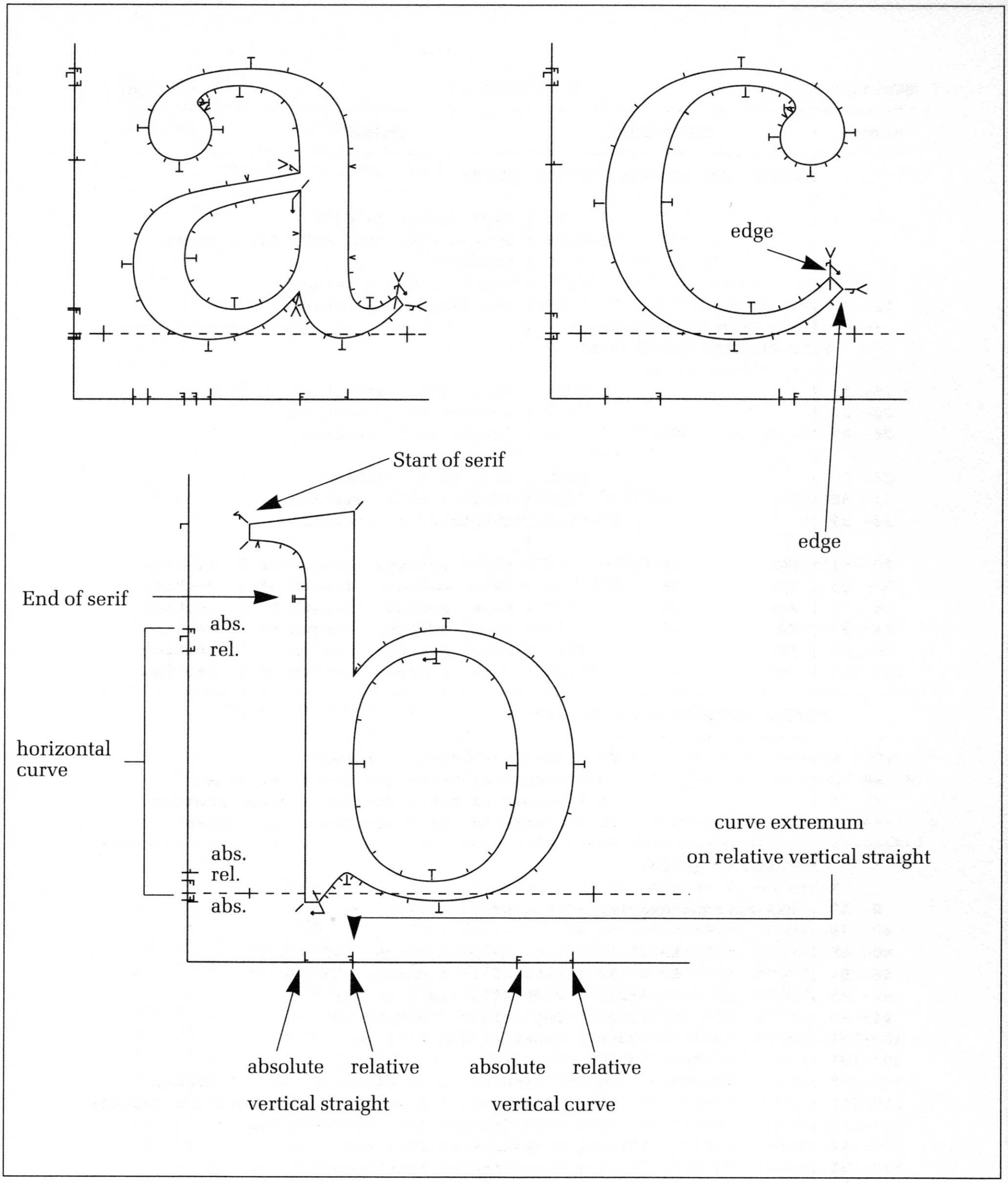
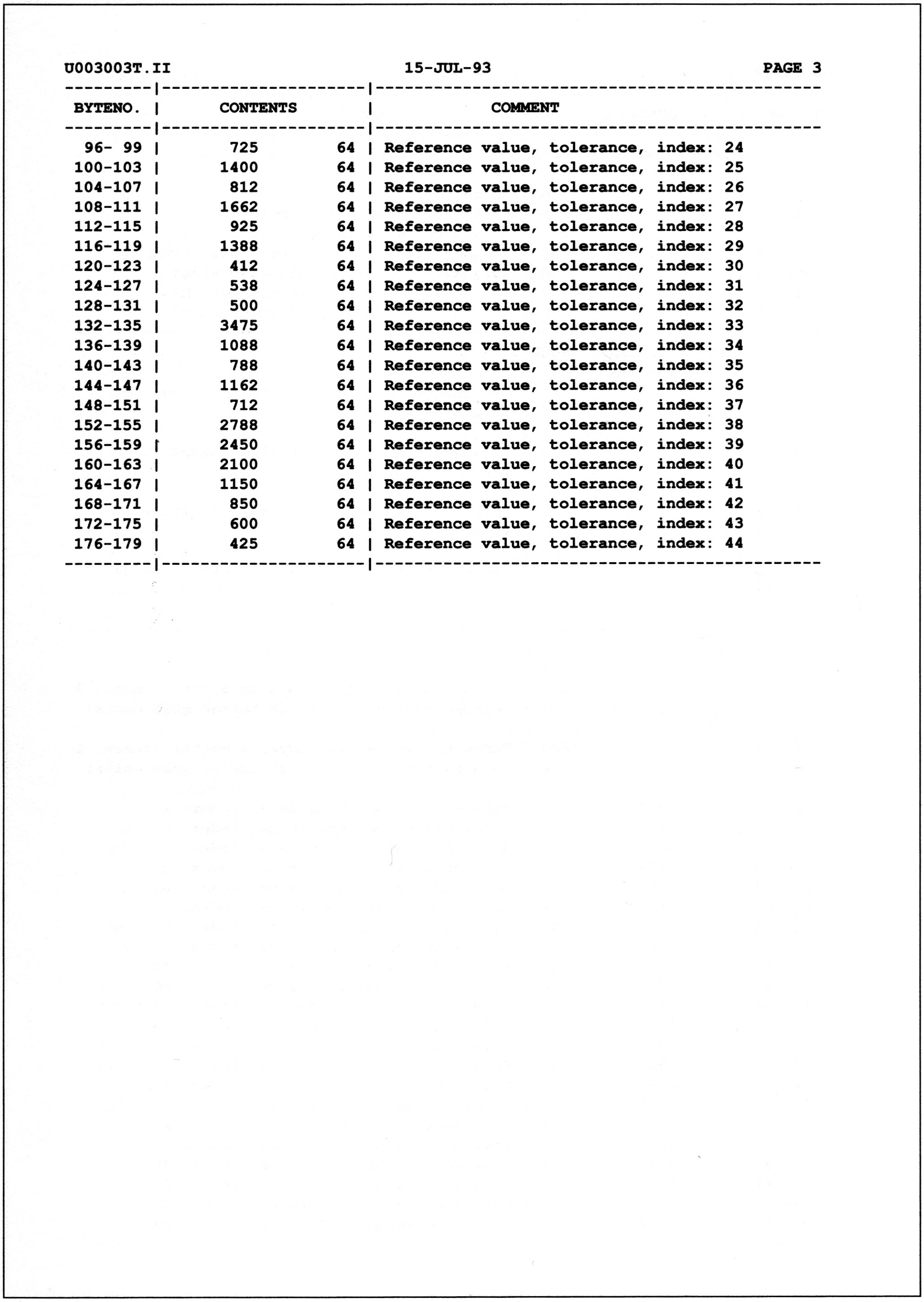

edge
edge
Start of serif
End of serif
abs.
rel.
horizontal curve
abs.
rel.
abs.
curve extremum
on relative vertical straight
absolute
relative
absolute
relative
vertical straight
vertical curve

Fig. 188
Images of
II format
characters

```
U003003T.II                        15-JUL-93                            PAGE 1
---------|--------------------------|--------------------------------------
BYTENO.  |         CONTENTS         |           COMMENT
---------|--------------------------|--------------------------------------
         CONTROL AND ADDRESS SECTION (CASE)
         ----------------------------------
   0-  1 |                       II | Data format (ASCII)
   2-  3 |                   0x0000 | Byte order: MAC, SUN, 68xxx family
   4-  7 |                        0 | Checksum
   8- 11 |                    96134 | Fontlength (in bytes)
  12- 15 |                       96 | FAB length (in bytes)
---------|--------------------------|--------------------------------------
        FILE ADDRESS BLOCK (FAB)
        ------------------------
  16- 19 |                     FDBL | Name of 1. section
  20- 23 |                      112 | Address of 1. section
  24- 27 |                        8 | Length of 1. section
         |                          |
  28- 31 |                     FOHD | Name of 2. section
  32- 35 |                      120 | Address of 2. section
  36- 39 |                      132 | Length of 2. section
         |                          |
  40- 51 | SCAL      252         16 | Name, address, length of 3. section
  52- 63 | FINF      268          0 | Name, address, length of 4. section
  64- 75 | FORT      268        180 | Name, address, length of 5. section
  76- 87 | FOGL      448         16 | Name, address, length of 6. section
  88- 99 | IKDR      464       3108 | Name, address, length of 7. section
 100-111 | IKDT     3572      92562 | Name, address, length of 8. section
---------|--------------------------|--------------------------------------
        FORMAT DEFINITION BLOCK (FDBL)
        ------------------------------
   0-  1 |                1.03 | Version number of format
   2-  3 |                   4 | Number of bytes for character names
   4-  5 |                   2 | Number of bytes for instruction indices
   6-  7 |                  28 | Length of character header (in bytes)
---------|---------------------|-------------------------------------------
         FONTHEADER (FOHD)
         -----------------
   0- 39 | URW Antiqua Regular | Font name (ASCII)
  40- 79 |                     |
  80- 85 |            15 07 93 | Date of production:  15-JUL-93
  86- 91 |            15 07 93 | Date of last change: 15-JUL-93
  92- 95 |               15000 | Body size (in 0.01 mm)
  96- 99 |               10000 | Cap height (in 0.01 mm)
 100-103 |                7000 | x-height (in 0.01 mm)
 104-107 |                3250 | Descender (in 0.01 mm)
 108-109 |                   0 | Italic angle of capitals (in 0.1 degree)
 110-111 |                   0 | Italic angle of small letters (in 0.1 degree)
 112-115 |                   0 | X-minimum of font bounding box
 116-119 |               15795 | X-maximum of font bounding box
 120-123 |               -3600 | Y-minimum of font bounding box
 124-127 |               13380 | Y-maximum of font bounding box
 128-131 |                 259 | Number of characters
---------|---------------------|-------------------------------------------
```

```
U003003T.II                       15-JUL-93                          PAGE 2
---------|--------------------|------------------------------------------
 BYTENO. |      CONTENTS      |               COMMENT
---------|--------------------|------------------------------------------
         FONTSCALING DATA (SCAL)
         -----------------------
   0-  3 |                  0 | Cache ID
   4-  5 |               1000 | X-resolution of target grid
   6-  7 |               1000 | Y-resolution of target grid
   8-  9 |               1300 | Canonic stem width (in bodysize units)
  10- 11 |                  0 | Tolerance of the canonic stem width
  12- 13 |                  0 | Cut in value of canonic stem width
  14- 15 |                  0 | Cut in value of dropout control
---------|--------------------|---------------------------------------------
         FONT GUIDE LINES (FOGL)
         -----------------------
   0-  1 |              10000 | Cap height (in bodysize units), index: 0
   2-  3 |                  0 | Tolerance value
         |                    |
   4-  5 |               7000 | X-height (in bodysize units), index: 1
   6-  7 |                  0 | Tolerance value
         |                    |
   8-  9 |                  0 | Baseline (in bodysize units), index: 2
  10- 11 |                  0 | Tolerance value
         |                    |
  12- 13 |              -3250 | Descender line (in bodysize units), index: 3
  14- 15 |                  0 | Tolerance value
         |                    |
---------|--------------------|---------------------------------------------
         FONT REFERENCE TABLE (FORT)
         ---------------------------
   0-  1 |                612 | Reference value (in bodysize units), index: 0
   2-  3 |                 64 | Tolerance value (in 1/128 target grid units)
         |                    |
   4-  5 |               1500 | Reference value (in bodysize units), index: 1
   6-  7 |                 64 | Tolerance value (in 1/128 target grid units)
         |                    |
   8- 11 |     1650        64 | Reference value, tolerance, index: 2
  12- 15 |      575        64 | Reference value, tolerance, index: 3
  16- 19 |     3125        64 | Reference value, tolerance, index: 4
  20- 23 |     1238        64 | Reference value, tolerance, index: 5
  24- 27 |     1375        64 | Reference value, tolerance, index: 6
  28- 31 |      888        64 | Reference value, tolerance, index: 7
  32- 35 |     3725        64 | Reference value, tolerance, index: 8
  36- 39 |     3200        64 | Reference value, tolerance, index: 9
  40- 43 |      350        64 | Reference value, tolerance, index: 10
  44- 47 |      638        64 | Reference value, tolerance, index: 11
  48- 51 |      450        64 | Reference value, tolerance, index: 12
  52- 55 |     1038        64 | Reference value, tolerance, index: 13
  56- 59 |     1312        64 | Reference value, tolerance, index: 14
  60- 63 |      800        64 | Reference value, tolerance, index: 15
  64- 67 |     1488        64 | Reference value, tolerance, index: 16
  68- 71 |      700        64 | Reference value, tolerance, index: 17
  72- 75 |      512        64 | Reference value, tolerance, index: 18
  76- 79 |     1000        64 | Reference value, tolerance, index: 19
  80- 83 |     1275        64 | Reference value, tolerance, index: 20
  84- 87 |     1112        64 | Reference value, tolerance, index: 21
  88- 91 |     1200        64 | Reference value, tolerance, index: 22
  92- 95 |      962        64 | Reference value, tolerance, index: 23
---------|--------------------|---------------------------------------------
```



```
U003003T.II                        15-JUL-93                            PAGE 3
---------|---------------------|---------------------------------------------
 BYTENO. |      CONTENTS       |             COMMENT
---------|---------------------|---------------------------------------------
  96- 99 |       725        64 | Reference value, tolerance, index: 24
 100-103 |      1400        64 | Reference value, tolerance, index: 25
 104-107 |       812        64 | Reference value, tolerance, index: 26
 108-111 |      1662        64 | Reference value, tolerance, index: 27
 112-115 |       925        64 | Reference value, tolerance, index: 28
 116-119 |      1388        64 | Reference value, tolerance, index: 29
 120-123 |       412        64 | Reference value, tolerance, index: 30
 124-127 |       538        64 | Reference value, tolerance, index: 31
 128-131 |       500        64 | Reference value, tolerance, index: 32
 132-135 |      3475        64 | Reference value, tolerance, index: 33
 136-139 |      1088        64 | Reference value, tolerance, index: 34
 140-143 |       788        64 | Reference value, tolerance, index: 35
 144-147 |      1162        64 | Reference value, tolerance, index: 36
 148-151 |       712        64 | Reference value, tolerance, index: 37
 152-155 |      2788        64 | Reference value, tolerance, index: 38
 156-159 |      2450        64 | Reference value, tolerance, index: 39
 160-163 |      2100        64 | Reference value, tolerance, index: 40
 164-167 |      1150        64 | Reference value, tolerance, index: 41
 168-171 |       850        64 | Reference value, tolerance, index: 42
 172-175 |       600        64 | Reference value, tolerance, index: 43
 176-179 |       425        64 | Reference value, tolerance, index: 44
---------|---------------------|---------------------------------------------
```

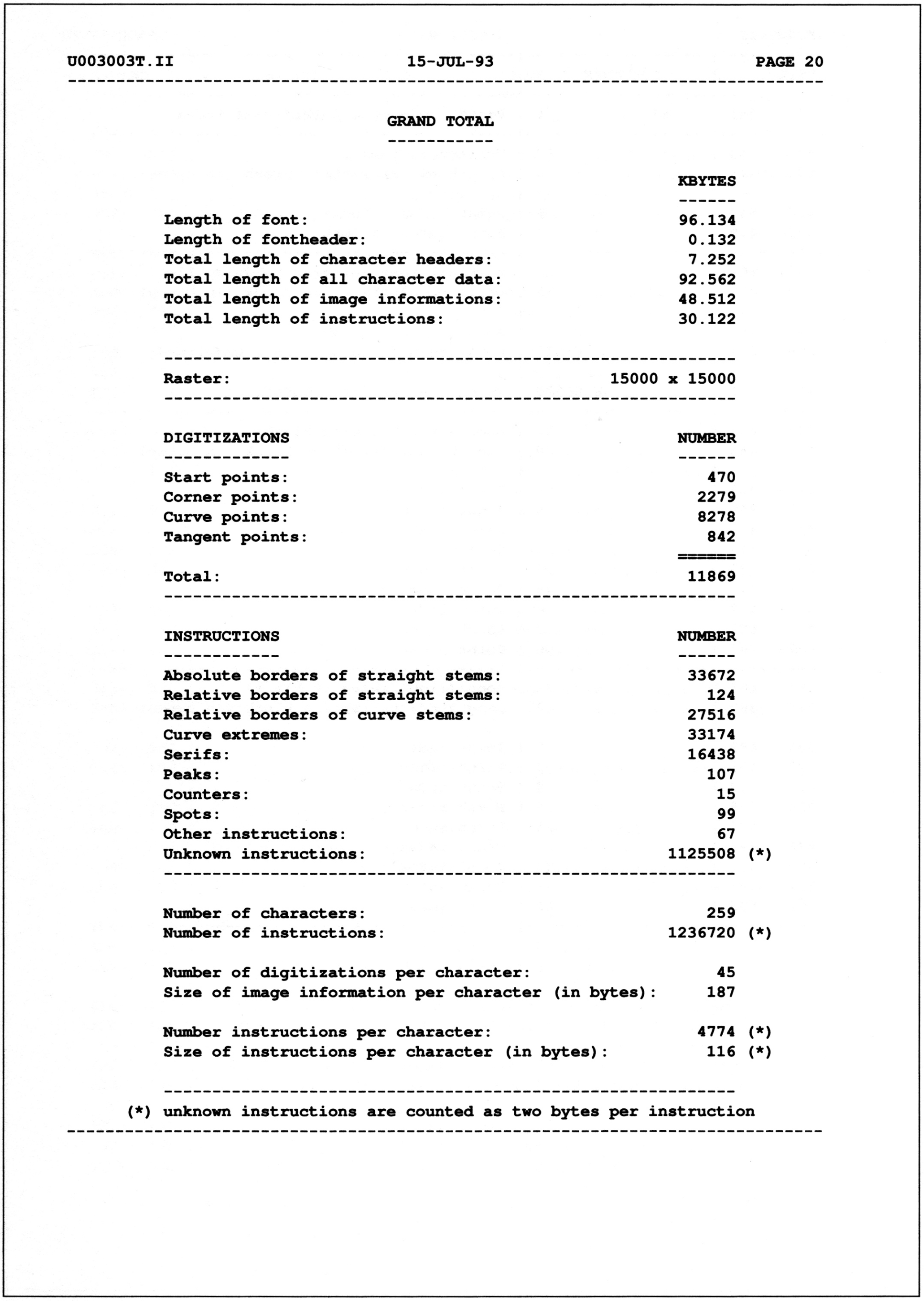
```
U003003T.II                     15-JUL-93                              PAGE 4
---------------|----------------------------|-------------------------------
   BYTENO.     |          CONTENTS          |        COMMENT
---------------|----------------------------|-------------------------------
                   IK DIRECTORY (IKDR)
                   -------------------
     0-    3 |                        101 | Character number;  char. 1
     4-    7 |                          0 | address relative IKDT-start
     8-   11 |                        348 | Length of data (in bytes)
             |                            |
    12-   15 |                        102 | Character number;  char. 2
    16-   19 |                        348 | address relative IKDT-start
    20-   23 |                        460 | Length of data (in bytes)
             |                            |
    24-   35 |        103       808   322 | No., address, length;       3
    36-   47 |        104      1130   324 | No., address, length;       4
    48-   59 |        105      1454   366 | No., address, length;       5
    60-   71 |        106      1820   340 | No., address, length;       6
    72-   83 |        107      2160   412 | No., address, length;       7
    84-   95 |        108      2572   450 | No., address, length;       8
    96-  107 |        109      3022   254 | No., address, length;       9
   108-  119 |        110      3276   250 | No., address, length;      10
   120-  131 |        111      3526   452 | No., address, length;      11
   132-  143 |        112      3978   280 | No., address, length;      12
   144-  155 |        113      4258   396 | No., address, length;      13
   156-  167 |        114      4654   328 | No., address, length;      14
   168-  179 |        115      4982   276 | No., address, length;      15
   180-  191 |        116      5258   362 | No., address, length;      16
   192-  203 |        117      5620   380 | No., address, length;      17
   204-  215 |        118      6000   522 | No., address, length;      18
   216-  227 |        119      6522   408 | No., address, length;      19
   228-  239 |        120      6930   262 | No., address, length;      20
   240-  251 |        121      7192   366 | No., address, length;      21
   252-  263 |        122      7558   282 | No., address, length;      22
   264-  275 |        123      7840   368 | No., address, length;      23
   276-  287 |        124      8208   502 | No., address, length;      24
   288-  299 |        125      8710   376 | No., address, length;      25
   300-  311 |        126      9086   234 | No., address, length;      26
   312-  323 |        127      9320   518 | No., address, length;      27
   324-  335 |        128      9838   496 | No., address, length;      28
   336-  347 |        129     10334   460 | No., address, length;      29
   348-  359 |        201     10794   540 | No., address, length;      30
   360-  371 |        202     11334   442 | No., address, length;      31
   372-  383 |        203     11776   442 | No., address, length;      32
   384-  395 |        204     12218   440 | No., address, length;      33
   396-  407 |        207     12658   544 | No., address, length;      34
   408-  419 |        208     13202   572 | No., address, length;      35
   420-  431 |        210     13774   474 | No., address, length;      36
   432-  443 |        215     14248   380 | No., address, length;      37
   444-  455 |        216     14628   558 | No., address, length;      38
---------------|----------------------------|-------------------------------
   660-  671 |        253     22530   468 | No., address, length;      56
   672-  683 |        254     22998   466 | No., address, length;      57
   684-  695 |        257     23464   470 | No., address, length;      58
   696-  707 |        259     23934   330 | No., address, length;      59
   708-  719 |        261     24264   406 | No., address, length;      60
   720-  731 |        268     24670   568 | No., address, length;      61
   732-  743 |        301     25238   506 | No., address, length;      62
   744-  755 |        302     25744   382 | No., address, length;      63
   756-  767 |        303     26126   342 | No., address, length;      64
---------------|----------------------------|-------------------------------
```

```
U003003T.II                        15-JUL-93                          PAGE 9
-------------|---------------|---------------------------------------------------
   BYTENO.   |   CONTENTS    |                COMMENT
-------------|---------------|---------------------------------------------------
             IK CHARACTER DATA (IKDT)
             -----------------------
             CHARACTER HEADER
             ----------------
    0-    3 |            301 | Character number
    4-    7 |            506 | Length of character data (in bytes)
    8-    9 |              0 | Control word
   10-   13 |             73 | Number of digitizations
   14-   15 |           8240 | Total width
   16-   17 |            799 | Left side bearing
   18-   19 |              0 | X-minimum
   20-   21 |           7298 | X-maximum
   22-   23 |           -150 | Y-minimum
   24-   25 |           7155 | Y-maximum
   26-   27 |              1 | Unit of measurement (in 0.01 mm)
-------------|---------------|---------------------------------------------------
             CONTOUR INDEX
             -------------
   28-   31 |             28 | Length of contour index (in bytes)
            |                |
   32-   35 |              0 | Index of starting point of 1. contour
   36-   37 |             -1 | Sense of rotation: clockwise
   38-   39 |              0 | Nesting: outer contour 1. order
   40-   41 |              0 | Colour: transparent
   42-   43 |             58 | Number of digitizations
            |                |
   44-   47 |             58 | Index of starting point of 2. contour
   48-   49 |              1 | Sense of rotation: counter clockwise
   50-   51 |              1 | Nesting: inner contour 1. order
   52-   53 |              0 | Colour: transparent
   54-   55 |             15 | Number of digitizations
-------------|---------------|---------------------------------------------------
             IMAGE INFORMATION
             -----------------
   56-   59 |            296 | Length of image information (in bytes)
            |                |
   60-   63 |  -7127    1185 | 12   7126   1034 (K,X,Y)     0.DIG, type: STRP
   64-   67 |  -7299    -942 | 13   7298    791 (K,X,Y)     1.DIG, type: -EP-
   68-   71 |   6929     597 | 14   6928    446 (K,X,Y)     2.DIG, type: .C..
   72-   75 |   6354     221 | 14   6353     70 (K,X,Y)     3.DIG, type: .C..
   76-   79 |   5641      46 | 14   5640   -105 (K,X,Y)     4.DIG, type: .C..
   80-   83 |   5216     144 | 14   5215     -7 (K,X,Y)     5.DIG, type: .C..
   84-   87 |   4791     491 | 14   4790    340 (K,X,Y)     6.DIG, type: .C..
   88-   91 |   4574     952 | 14   4573    801 (K,X,Y)     7.DIG, type: .C..
   92-   95 |  -4488   -1285 | 13   4487   1134 (K,X,Y)     8.DIG, type: -EP-
   96-   99 |   4069     842 | 14   4068    691 (K,X,Y)     9.DIG, type: .C..
  100-  103 |   3488     406 | 14   3487    255 (K,X,Y)    10.DIG, type: .C..
  104-  107 |   2724      88 | 14   2723    -63 (K,X,Y)    11.DIG, type: .C..
  108-  111 |   2035       1 | 14   2034   -150 (K,X,Y)    12.DIG, type: .C..
  112-  115 |    992     239 | 14    991     88 (K,X,Y)    13.DIG, type: .C..
  116-  119 |    197    1069 | 14    196    918 (K,X,Y)    14.DIG, type: .C..
  120-  123 |      1    2034 | 14      0   1883 (K,X,Y)    15.DIG, type: .C..
  124-  127 |    139    2822 | 14    138   2671 (K,X,Y)    16.DIG, type: .C..
  128-  131 |    543    3419 | 14    542   3268 (K,X,Y)    17.DIG, type: .C..
  132-  135 |   1532    3943 | 14   1531   3792 (K,X,Y)    18.DIG, type: .C..
-------------|---------------|---------------------------------------------------
```

```
U003003T.II                        15-JUL-93                         PAGE 10
-------------|---------------|----------------------------------------------
   BYTENO.   |   CONTENTS    |              COMMENT
-------------|---------------|----------------------------------------------
   136-  139 |   2526   4150 | 14   2525   3999 (K,X,Y)     19.DIG, type: .C..
   140-  143 |   3097  -4248 | 15   3096   4097 (K,X,Y)     20.DIG, type: >TP<
   144-  147 |  -4494  -4500 | 13   4493   4349 (K,X,Y)     21.DIG, type: -EP-
   148-  151 |   4479   5105 | 14   4478   4954 (K,X,Y)     22.DIG, type: .C..
   152-  155 |   4417   5719 | 14   4416   5568 (K,X,Y)     23.DIG, type: .C..
   156-  159 |   4072   6463 | 14   4071   6312 (K,X,Y)     24.DIG, type: .C..
   160-  163 |   3471   6782 | 14   3470   6631 (K,X,Y)     25.DIG, type: .C..
   164-  167 |   2858   6856 | 14   2857   6705 (K,X,Y)     26.DIG, type: .C..
   168-  171 |   2181   6755 | 14   2180   6604 (K,X,Y)     27.DIG, type: .C..
   172-  175 |   1945   6642 | 14   1944   6491 (K,X,Y)     28.DIG, type: .C..
   176-  179 |   1805   6531 | 14   1804   6380 (K,X,Y)     29.DIG, type: .C..
   180-  183 |   1726   6394 | 14   1725   6243 (K,X,Y)     30.DIG, type: .C..
   184-  187 |   1779   6290 | 14   1778   6139 (K,X,Y)     31.DIG, type: .C..
   188-  191 |   1907   6163 | 14   1906   6012 (K,X,Y)     32.DIG, type: .C..
   192-  195 |   2051   5936 | 14   2050   5785 (K,X,Y)     33.DIG, type: .C..
   196-  199 |   2101   5660 | 14   2100   5509 (K,X,Y)     34.DIG, type: .C..
   200-  203 |   2004   5267 | 14   2003   5116 (K,X,Y)     35.DIG, type: .C..
   204-  207 |   1680   4933 | 14   1679   4782 (K,X,Y)     36.DIG, type: .C..
   208-  211 |   1237   4831 | 14   1236   4680 (K,X,Y)     37.DIG, type: .C..
   212-  215 |    875   4902 | 14    874   4751 (K,X,Y)     38.DIG, type: .C..
   216-  219 |    489   5307 | 14    488   5156 (K,X,Y)     39.DIG, type: .C..
   220-  223 |    406   5704 | 14    405   5553 (K,X,Y)     40.DIG, type: .C..
   224-  227 |    502   6163 | 14    501   6012 (K,X,Y)     41.DIG, type: .C..
   228-  231 |    966   6753 | 14    965   6602 (K,X,Y)     42.DIG, type: .C..
   232-  235 |   1905   7170 | 14   1904   7019 (K,X,Y)     43.DIG, type: .C..
   236-  239 |   3171   7306 | 14   3170   7155 (K,X,Y)     44.DIG, type: .C..
   240-  243 |   4151   7215 | 14   4150   7064 (K,X,Y)     45.DIG, type: .C..
   244-  247 |   5042   6837 | 14   5041   6686 (K,X,Y)     46.DIG, type: .C..
   248-  251 |   5653   6035 | 14   5652   5884 (K,X,Y)     47.DIG, type: .C..
   252-  255 |   5815   5153 | 14   5814   5002 (K,X,Y)     48.DIG, type: .C..
   256-  259 |   5821  -4680 | 15   5820   4529 (K,X,Y)     49.DIG, type: >TP<
   260-  263 |   5821  -2223 | 15   5820   2072 (K,X,Y)     50.DIG, type: >TP<
   264-  267 |   5827   1813 | 14   5826   1662 (K,X,Y)     51.DIG, type: .C..
   268-  271 |   5877   1345 | 14   5876   1194 (K,X,Y)     52.DIG, type: .C..
   272-  275 |   6079    958 | 14   6078    807 (K,X,Y)     53.DIG, type: .C..
   276-  279 |   6392    841 | 14   6391    690 (K,X,Y)     54.DIG, type: .C..
   280-  283 |   6755    938 | 14   6754    787 (K,X,Y)     55.DIG, type: .C..
   284-  287 |   6979   1081 | 14   6978    930 (K,X,Y)     56.DIG, type: .C..
   288-  291 |  -7127  -1185 | 13   7126   1034 (K,X,Y)     57.DIG, type: -EP-
   292-  295 |  -4516   4038 | 12   4515   3887 (K,X,Y)     58.DIG, type: STRP
   296-  299 |   3231   3793 | 14   3230   3642 (K,X,Y)     59.DIG, type: .C..
   300-  303 |   2756   3678 | 14   2755   3527 (K,X,Y)     60.DIG, type: .C..
   304-  307 |   2082   3440 | 14   2081   3289 (K,X,Y)     61.DIG, type: .C..
   308-  311 |   1562   2936 | 14   1561   2785 (K,X,Y)     62.DIG, type: .C..
   312-  315 |   1396   2200 | 14   1395   2049 (K,X,Y)     63.DIG, type: .C..
   316-  319 |   1528   1556 | 14   1527   1405 (K,X,Y)     64.DIG, type: .C..
   320-  323 |   1996   1027 | 14   1995    876 (K,X,Y)     65.DIG, type: .C..
   324-  327 |   2747    826 | 14   2746    675 (K,X,Y)     66.DIG, type: .C..
   328-  331 |   3466    978 | 14   3465    827 (K,X,Y)     67.DIG, type: .C..
   332-  335 |   4060   1391 | 14   4059   1240 (K,X,Y)     68.DIG, type: .C..
   336-  339 |   4354   1835 | 14   4353   1684 (K,X,Y)     69.DIG, type: .C..
   340-  343 |   4491   2436 | 14   4490   2285 (K,X,Y)     70.DIG, type: .C..
   344-  347 |   4516  -2868 | 15   4515   2717 (K,X,Y)     71.DIG, type: >TP<
   348-  351 |  -4516  -4038 | 13   4515   3887 (K,X,Y)     72.DIG, type: -EP-
-------------|---------------|----------------------------------------------
```

```
U003003T.II                        15-JUL-93                           PAGE 11
-------------|-------------------|--------------------------------------------
   BYTENO.   |     CONTENTS      |           COMMENT
-------------|-------------------|--------------------------------------------
                INSTRUCTIONS
                ------------
   352-  355 |               154 | Length of instructions (in bytes)
-------------|-------------------|--------------------------------------------
   356-  357 |                 2 | Absolute vertical line
   358-  359 |                 6 | Length of instruction block (in bytes)
             |                   |
   360-  361 |                49 | Point index
-------------|-------------------|--------------------------------------------
   362-  363 |                 5 | Relative vertical stem border
   364-  365 |                10 | Length of instruction block (in bytes)
             |                   |
   366-  367 |                71 | Point index
   368-  369 |                49 | Reference point index
   370-  371 |                14 | FORT index of stem width
             |                   |
-------------|-------------------|--------------------------------------------
   372-  373 |                 6 | Horizontal curve extremum
   374-  375 |                 6 | Length of instruction block (in bytes)
             |                   |
   376-  377 |                37 | Point index of extremum
-------------|-------------------|--------------------------------------------
   378-  379 |                 7 | Vertical curve extremum
   380-  381 |                10 | Length of instruction block (in bytes)
             |                   |
   382-  383 |                15 | Point index of extremum
   384-  385 |                34 | Point index of extremum
   386-  387 |                40 | Point index of extremum
-------------|-------------------|--------------------------------------------
   388-  389 |                 8 | Horizontal curve referred to guide line
   390-  391 |                16 | Length of instruction block (in bytes)
             |                   |
   392-  393 |                 4 | Point index
   394-  395 |                 2 | FOGL index of guide line : baseline
             |                   |
   396-  399 |          12     2 | Point, guide line : baseline
   400-  403 |          44     1 | Point, guide line : x-height
-------------|-------------------|--------------------------------------------
   404-  405 |                 9 | Relative horizontal curve stem border
   406-  407 |                22 | Length of instruction block (in bytes)
             |                   |
   408-  409 |                26 | Point index
   410-  411 |                44 | Reference point index
   412-  413 |                12 | FORT index of stem width
             |                   |
   414-  419 |    54     4    15 | Point, reference point, FORT index
   420-  425 |    66    12    15 | Point, reference point, FORT index
-------------|-------------------|--------------------------------------------
   426-  427 |                10 | Relative vertical curve stem border
   428-  429 |                16 | Length of instruction block (in bytes)
             |                   |
   430-  431 |                30 | Point index
   432-  433 |                40 | Reference point index
   434-  435 |                 6 | FORT index of stem width
             |                   |
-------------|-------------------|--------------------------------------------
```

```
U003003T.II                          15-JUL-93                          PAGE 12
-------------|--------------------|-----------------------------------------
   BYTENO.   |      CONTENTS      |            COMMENT
-------------|--------------------|-----------------------------------------
   436-  441 |   63    15       6 | Point, reference point, FORT index
-------------|--------------------|-----------------------------------------
   442-  443 |                 13 | Horizontal peak
   444-  445 |                  8 | Length of instruction block (in bytes)
             |                    |
   446-  447 |                  1 | Point index of peak
   448-  449 |                 21 | Point index of peak
-------------|--------------------|-----------------------------------------
   450-  451 |                 14 | Vertical peak referred to guide line
   452-  453 |                 12 | Length of instruction block (in bytes)
             |                    |
   454-  455 |                  0 | Point index of peak
   456-  457 |              65535 | FOGL index of guide line : undefined
             |                    |
   458-  461 |            8 65535 | Point, guide line : undefined
-------------|--------------------|-----------------------------------------
   462-  463 |                 20 | Spot (left/below is black)
   464-  465 |                 22 | Length of instruction block (in bytes)
             |                    |
   466-  467 |                 49 | Point index
   468-  469 |                  0 | Point index
   470-  471 |                  1 | Point index
   472-  473 |                 30 | Point index
   474-  475 |                 34 | Point index
   476-  477 |                 44 | Point index
   478-  479 |                 54 | Point index
   480-  481 |                 63 | Point index
   482-  483 |                 66 | Point index
-------------|--------------------|-----------------------------------------
   484-  485 |                 21 | Spot (right/above is black)
   486-  487 |                 22 | Length of instruction block (in bytes)
             |                    |
   488-  489 |                 71 | Point index
   490-  491 |                  4 | Point index
   492-  493 |                  8 | Point index
   494-  495 |                 12 | Point index
   496-  497 |                 15 | Point index
   498-  499 |                 21 | Point index
   500-  501 |                 26 | Point index
   502-  503 |                 37 | Point index
   504-  505 |                 40 | Point index
-------------|--------------------|-----------------------------------------
```



```
U003003T.II                        15-JUL-93                            PAGE 20
-------------------------------------------------------------------------------

                                  GRAND TOTAL
                                  -----------

                                                             KBYTES
                                                             ------
          Length of font:                                    96.134
          Length of fontheader:                               0.132
          Total length of character headers:                  7.252
          Total length of all character data:                92.562
          Total length of image informations:                48.512
          Total length of instructions:                      30.122

          -----------------------------------------------------------
          Raster:                                     15000 x 15000
          -----------------------------------------------------------

          DIGITIZATIONS                                      NUMBER
          -------------                                      ------
          Start points:                                         470
          Corner points:                                       2279
          Curve points:                                        8278
          Tangent points:                                       842
                                                             ======
          Total:                                              11869
          -----------------------------------------------------------

          INSTRUCTIONS                                       NUMBER
          ------------                                       ------
          Absolute borders of straight stems:                 33672
          Relative borders of straight stems:                   124
          Relative borders of curve stems:                    27516
          Curve extremes:                                     33174
          Serifs:                                             16438
          Peaks:                                                107
          Counters:                                              15
          Spots:                                                 99
          Other instructions:                                    67
          Unknown instructions:                             1125508 (*)
          -----------------------------------------------------------

          Number of characters:                                 259
          Number of instructions:                           1236720 (*)

          Number of digitizations per character:                 45
          Size of image information per character (in bytes):   187

          Number instructions per character:                   4774 (*)
          Size of instructions per character (in bytes):        116 (*)

          -----------------------------------------------------------
      (*) unknown instructions are counted as two bytes per instruction
-------------------------------------------------------------------------------
```

Description of the DI format

1. Structure of a DI font

The structure of a DI font is as described in Appendix G.

2. Description of the DI format

Because most graphic display terminals and NC machines (plotters) have hardware and software to decode and raster straight and circular line commands given by the host computer, we have created a general, **binary display** format.

This is a closed contour format (as is the IK format) which describes contours of any arbitrary flat surface as a sequence of straights and arcs (curves).

With regard to typesetters using circular and straight line interpolation, and for drawing machines addressing only a certain grid resolution with line commands, the DI format may also be supplied so that all positions of straights and arcs match points within the corresponding equipment grid. This display format describes a contour such that it contains the coordinates of the end points of all straights and arcs, and radii and senses of arc rotation in a binary form. Special command codes for various coding of certain machine types (e.g., in ASCII) can be easily derived by use of a simple converter program (a post processor). Such post processors needs to be programmed with only simple formulae from elementary geometry, in order to be able to generate other determinant arc parameters for these machine types from data supplied (e.g. instead of end point and radius: angle interval, start angle and direction of tangent).

Each character has its own coordinate system with the X zero point at the X minimum of the character, and the Y zero point on the base line.

Straights are described using three words (K, X, Y) in which the identification K = 100 applies to non-visible straights (movement of the writing head without actually writing) and K = 200 applying to visible straights. Circles are described in 5 words (K, X, Y, R) where K = 300. X and Y are the absolute X and Y positions of the circle segment end points in the character coordinate system. R is the radius of the circle, in stored as a long integer.

The sign of the R gives the direction (sense) of rotation of the circle: positive R means anti-clockwise direction (mathematically right).

The DI format is calculated from the IK format. For the purposes of easy reconversion, the identification K of a point which was also an IK point is raised by the IK identification. Thus, DI identifications are:

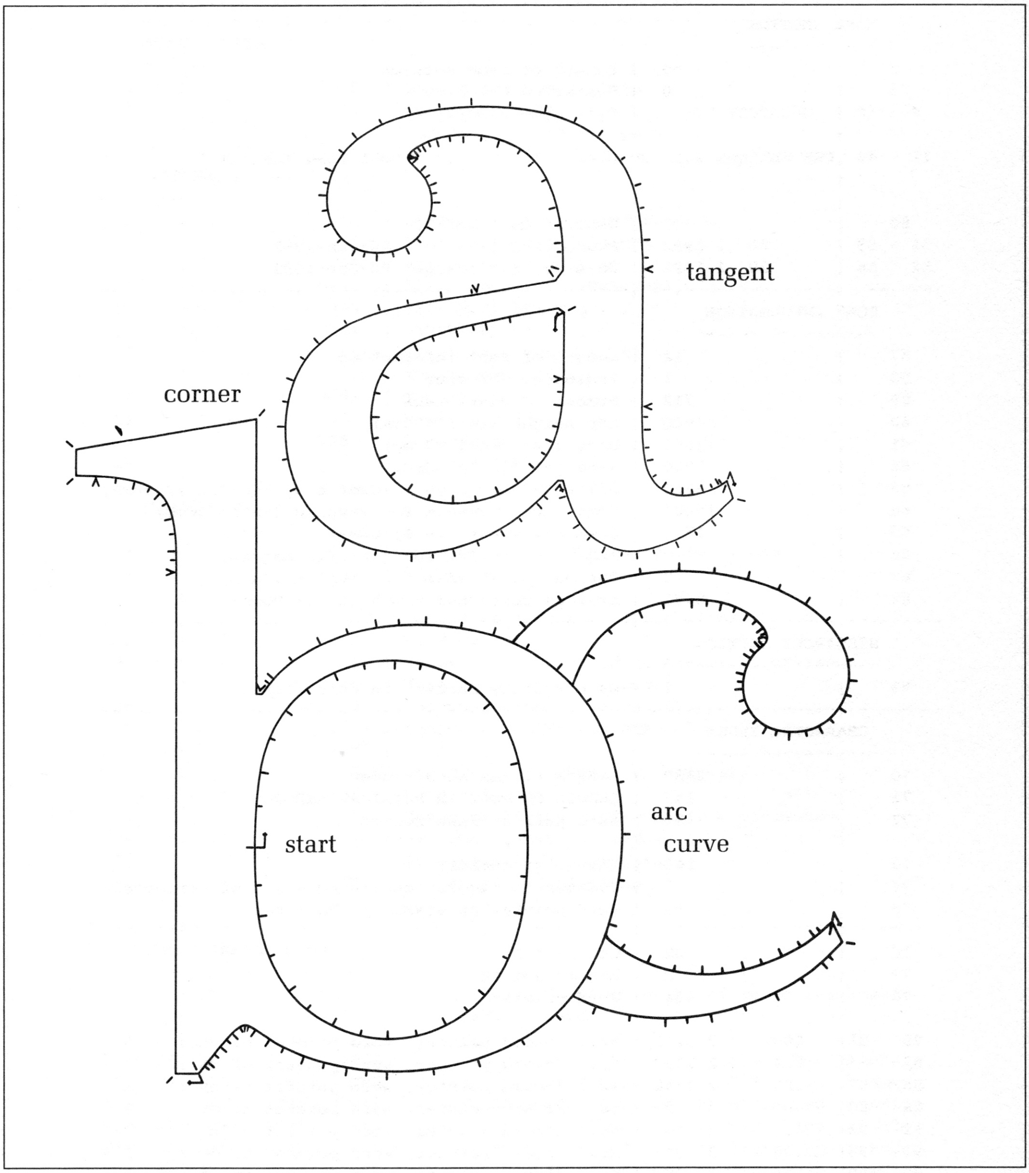
K = 112 for starting points,
K = 200, 213, 214, 215 for straight end points,
K = 300, 313, 314, 315 for arc segment end points.

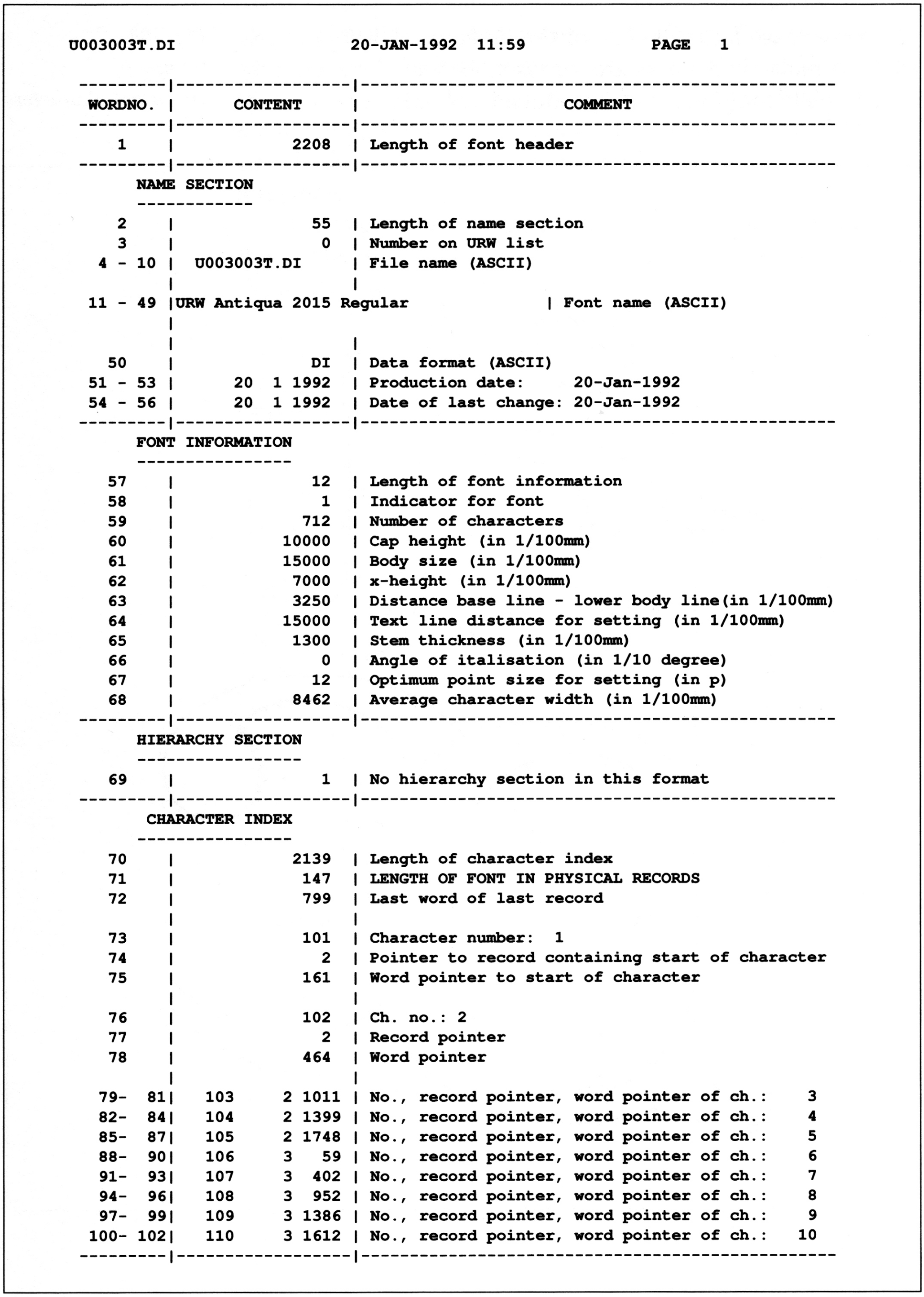
The cap height is set to the standard value of 100 mm unless otherwise provided. Each straight and each arc is counted as one digitization.

3. Example of a DI format

The lower case letters a, b, c of the typeface «URW Antiqua 2015 normal», with marks are represented first. Subsequently the complete font format is reproduced as a data list.

Fig. 189
Images of
DI format characters


tangent
corner
start
arc
curve



```
U003003T.DI                  20-JAN-1992  11:59             PAGE   1

 ---------|------------------|-------------------------------------------------
  WORDNO. |      CONTENT     |                     COMMENT
 ---------|------------------|-------------------------------------------------
     1    |            2208  | Length of font header
 ---------|------------------|-------------------------------------------------
       NAME SECTION
       ------------
     2    |              55  | Length of name section
     3    |               0  | Number on URW list
   4 - 10 |  U003003T.DI     | File name (ASCII)
          |                  |
  11 - 49 |URW Antiqua 2015 Regular              | Font name (ASCII)
          |
          |                  |
    50    |              DI  | Data format (ASCII)
  51 - 53 |      20  1 1992  | Production date:     20-Jan-1992
  54 - 56 |      20  1 1992  | Date of last change: 20-Jan-1992
 ---------|------------------|-------------------------------------------------
       FONT INFORMATION
       ----------------
    57    |              12  | Length of font information
    58    |               1  | Indicator for font
    59    |             712  | Number of characters
    60    |           10000  | Cap height (in 1/100mm)
    61    |           15000  | Body size (in 1/100mm)
    62    |            7000  | x-height (in 1/100mm)
    63    |            3250  | Distance base line - lower body line(in 1/100mm)
    64    |           15000  | Text line distance for setting (in 1/100mm)
    65    |            1300  | Stem thickness (in 1/100mm)
    66    |               0  | Angle of italisation (in 1/10 degree)
    67    |              12  | Optimum point size for setting (in p)
    68    |            8462  | Average character width (in 1/100mm)
 ---------|------------------|-------------------------------------------------
       HIERARCHY SECTION
       -----------------
    69    |               1  | No hierarchy section in this format
 ---------|------------------|-------------------------------------------------
        CHARACTER INDEX
       ----------------
    70    |            2139  | Length of character index
    71    |             147  | LENGTH OF FONT IN PHYSICAL RECORDS
    72    |             799  | Last word of last record
          |                  |
    73    |             101  | Character number:  1
    74    |               2  | Pointer to record containing start of character
    75    |             161  | Word pointer to start of character
          |                  |
    76    |             102  | Ch. no.: 2
    77    |               2  | Record pointer
    78    |             464  | Word pointer
          |                  |
   79-  81|   103     2 1011 | No., record pointer, word pointer of ch.:    3
   82-  84|   104     2 1399 | No., record pointer, word pointer of ch.:    4
   85-  87|   105     2 1748 | No., record pointer, word pointer of ch.:    5
   88-  90|   106     3   59 | No., record pointer, word pointer of ch.:    6
   91-  93|   107     3  402 | No., record pointer, word pointer of ch.:    7
   94-  96|   108     3  952 | No., record pointer, word pointer of ch.:    8
   97-  99|   109     3 1386 | No., record pointer, word pointer of ch.:    9
  100- 102|   110     3 1612 | No., record pointer, word pointer of ch.:   10
 ---------|------------------|-------------------------------------------------
```

```
U003003T.DI                  20-JAN-1992  12:00           PAGE  15

 ---------|------------------|------------------------------------------------
 WORDNO.  |      CONTENT     |                    COMMENT
 ---------|------------------|------------------------------------------------
     1    |                 0 | Records      length of character
     2    |               714 | Words
 ---------|------------------|------------------------------------------------
       NAME SECTION
       ------------
     3    |                 3 | Length of name section
     4    |               301 | Character number
     5    |                 0 | Number of following part
 ---------|------------------|------------------------------------------------
       SETTING INFORMATION
       -------------------
     6    |                12 | Length of setting information
     7    |                 1 | Character type = letter
     8    |               139 | Number of DI-points
     9    |              8240 | Total set width (T=L+W+R)
    10    |               799 | Left side bearing (L)
    11    |              7291 | Width (W)
    12    |               150 | Right side bearing (R)
    13    |                 0 | X-minimum
    14    |              7291 | X-maximum
    15    |              -150 | Y-minimum
    16    |              7153 | Y-maximum
    17    |                 1 | Unit is 1/100mm
 ---------|------------------|------------------------------------------------
       CONTOUR INDEX
       -------------
    18    |                 0 | Records     length of contour index
    19    |                14 | Words
          |                   |
    20    |                 0 | Record pointer for   1. contour:     0
    21    |                 3 | Word pointer                   :    34
    22    |                -1 | Sense of rotation of contour = clockwise
    23    |                 0 | Nesting: outer  1. order
    24    |                 0 | Colour inside: transparent
    25    |               111 | Number of digitizations
          |                   |
    26    |                 0 | Record pointer for   2. contour:     0
    27    |               548 | Word pointer                   :   579
    28    |                -1 | Sense of rotation of contour = clockwise
    29    |                 1 | Nesting: inner  1. order
    30    |                 0 | Colour inside: transparent
    31    |                28 | Number of digitizations
          |                   |
 ---------|------------------|------------------------------------------------
       IMAGE INFORMATION
       -----------------
    32    |                             0 | Records lenght of image information
    33    |                           683 | Words
          |                               |
   34-  36| 112    7119   1033           | (K,X,Y);       1. DIG;  typ: STRP
   37-  39| 213    7291    790           | (K,X,Y);       2. DIG;  typ: vector
   40-  44| 300    7084    593    -11209 | (K,X,Y,R);     3. DIG;  typ: circle
   45-  49| 314    6921    447     -6611 | (K,X,Y,R);     4. DIG;  typ: circle
   50-  54| 300    6648    241     -2509 | (K,X,Y,R);     5. DIG;  typ: circle
   55-  59| 314    6346     72     -2563 | (K,X,Y,R);     6. DIG;  typ: circle
   60-  64| 300    5981    -58     -1951 | (K,X,Y,R);     7. DIG;  typ: circle
 ---------|-----------------------------|------------------------------------
```

```
     U003003T.DI                 20-JAN-1992  12:00           PAGE  16

      ---------|-------------------------------|-----------------------------------
       WORDNO. |            CONTENT            |              COMMENT
      ---------|-------------------------------|-----------------------------------
        65-  69| 314    5647    -102      -1476 | (K,X,Y,R);    8. DIG;  typ: circle
        70-  74| 300    5436     -80       -900 | (K,X,Y,R);    9. DIG;  typ: circle
        75-  79| 314    5210      -4      -1145 | (K,X,Y,R);   10. DIG;  typ: circle
        80-  84| 300    4966     148      -1125 | (K,X,Y,R);   11. DIG;  typ: circle
        85-  89| 314    4787     342       -952 | (K,X,Y,R);   12. DIG;  typ: circle
        90-  94| 300    4663     556      -1484 | (K,X,Y,R);   13. DIG;  typ: circle
        95-  99| 314    4571     801      -1659 | (K,X,Y,R);   14. DIG;  typ: circle
       100- 104| 300    4532     945      -5233 | (K,X,Y,R);   15. DIG;  typ: circle
       105- 109| 313    4485    1133      -8805 | (K,X,Y,R);   16. DIG;  typ: circle
       110- 114| 300    4240     867     -12207 | (K,X,Y,R);   17. DIG;  typ: circle
       115- 119| 314    4066     691      -5724 | (K,X,Y,R);   18. DIG;  typ: circle
       120- 124| 300    3804     468      -2752 | (K,X,Y,R);   19. DIG;  typ: circle
       125- 129| 314    3485     255      -3423 | (K,X,Y,R);   20. DIG;  typ: circle
       130- 134| 300    3106      66      -3102 | (K,X,Y,R);   21. DIG;  typ: circle
       135- 139| 314    2722     -63      -2830 | (K,X,Y,R);   22. DIG;  typ: circle
       140- 144| 300    2381    -128      -2758 | (K,X,Y,R);   23. DIG;  typ: circle
       145- 149| 314    2034    -150      -2772 | (K,X,Y,R);   24. DIG;  typ: circle
       150- 154| 300    1475     -87      -2510 | (K,X,Y,R);   25. DIG;  typ: circle
       155- 159| 314     991      88      -2103 | (K,X,Y,R);   26. DIG;  typ: circle
       160- 164| 300     516     440      -1765 | (K,X,Y,R);   27. DIG;  typ: circle
       165- 169| 314     196     917      -1666 | (K,X,Y,R);   28. DIG;  typ: circle
       170- 174| 300      51    1369      -2206 | (K,X,Y,R);   29. DIG;  typ: circle
       175- 179| 314       0    1873      -2504 | (K,X,Y,R);   30. DIG;  typ: circle
       180- 184| 300      37    2298      -2490 | (K,X,Y,R);   31. DIG;  typ: circle
       185- 189| 314     138    2669      -2030 | (K,X,Y,R);   32. DIG;  typ: circle
       190- 194| 300     314    3008      -1642 | (K,X,Y,R);   33. DIG;  typ: circle
       195- 199| 314     542    3266      -1329 | (K,X,Y,R);   34. DIG;  typ: circle
       200- 204| 300    1005    3584      -2451 | (K,X,Y,R);   35. DIG;  typ: circle
       205- 209| 314    1530    3789      -2477 | (K,X,Y,R);   36. DIG;  typ: circle
       210- 214| 300    1957    3890      -8266 | (K,X,Y,R);   37. DIG;  typ: circle
       215- 219| 314    2523    3996     -14220 | (K,X,Y,R);   38. DIG;  typ: circle
       220- 224| 300    2884    4057      73968 | (K,X,Y,R);   39. DIG;  typ: circle
       225- 229| 315    3094    4094      25144 | (K,X,Y,R);   40. DIG;  typ: circle
       230- 232| 213    4491    4346            | (K,X,Y);     41. DIG;  typ: vector
       233- 237| 300    4484    4688      42604 | (K,X,Y,R);   42. DIG;  typ: circle
       238- 242| 314    4476    4950      25034 | (K,X,Y,R);   43. DIG;  typ: circle
       243- 247| 300    4454    5275       4981 | (K,X,Y,R);   44. DIG;  typ: circle
       248- 252| 314    4414    5563       3944 | (K,X,Y,R);   45. DIG;  typ: circle
       253- 257| 300    4272    6015       1825 | (K,X,Y,R);   46. DIG;  typ: circle
       258- 262| 314    4069    6306       1025 | (K,X,Y,R);   47. DIG;  typ: circle
       263- 267| 300    3823    6488       1016 | (K,X,Y,R);   48. DIG;  typ: circle
       268- 272| 314    3467    6624       1581 | (K,X,Y,R);   49. DIG;  typ: circle
       273- 277| 300    3168    6679       2476 | (K,X,Y,R);   50. DIG;  typ: circle
       278- 282| 314    2859    6698       2555 | (K,X,Y,R);   51. DIG;  typ: circle
       283- 287| 300    2491    6671       2512 | (K,X,Y,R);   52. DIG;  typ: circle
       288- 292| 314    2176    6597       1942 | (K,X,Y,R);   53. DIG;  typ: circle
       293- 297| 300    2063    6552        916 | (K,X,Y,R);   54. DIG;  typ: circle
       298- 302| 314    1940    6484       1228 | (K,X,Y,R);   55. DIG;  typ: circle
       303- 307| 300    1867    6432        855 | (K,X,Y,R);   56. DIG;  typ: circle
       308- 312| 314    1800    6374        838 | (K,X,Y,R);   57. DIG;  typ: circle
       313- 317| 300    1736    6292        393 | (K,X,Y,R);   58. DIG;  typ: circle
       318- 322| 314    1721    6240        105 | (K,X,Y,R);   59. DIG;  typ: circle
       323- 327| 300    1732    6191        105 | (K,X,Y,R);   60. DIG;  typ: circle
       328- 332| 314    1775    6133        220 | (K,X,Y,R);   61. DIG;  typ: circle
       333- 335| 214    1905    6006            | (K,X,Y);     62. DIG;  typ: vector
       336- 340| 300    1988    5905       -596 | (K,X,Y,R);   63. DIG;  typ: circle
      ---------|-------------------------------|-----------------------------------
```

```
U003003T.DI                    20-JAN-1992  12:00           PAGE  17

 ---------|-------------------------------|-----------------------------------
  WORDNO. |            CONTENT            |              COMMENT
 ---------|-------------------------------|-----------------------------------
  341- 345| 314   2052   5780         -680 | (K,X,Y,R);   64. DIG;  typ: circle
  346- 350| 300   2090   5646         -743 | (K,X,Y,R);   65. DIG;  typ: circle
  351- 355| 314   2103   5504         -781 | (K,X,Y,R);   66. DIG;  typ: circle
  356- 360| 300   2078   5301         -844 | (K,X,Y,R);   67. DIG;  typ: circle
  361- 365| 314   2006   5112         -834 | (K,X,Y,R);   68. DIG;  typ: circle
  366- 370| 300   1861   4913         -863 | (K,X,Y,R);   69. DIG;  typ: circle
  371- 375| 314   1681   4779         -712 | (K,X,Y,R);   70. DIG;  typ: circle
  376- 380| 300   1479   4704         -843 | (K,X,Y,R);   71. DIG;  typ: circle
  381- 385| 314   1236   4677        -1086 | (K,X,Y,R);   72. DIG;  typ: circle
  386- 390| 300   1031   4697        -1070 | (K,X,Y,R);   73. DIG;  typ: circle
  391- 395| 314    873   4748         -694 | (K,X,Y,R);   74. DIG;  typ: circle
  396- 400| 300    650   4907         -738 | (K,X,Y,R);   75. DIG;  typ: circle
  401- 405| 314    485   5153         -857 | (K,X,Y,R);   76. DIG;  typ: circle
  406- 410| 300    423   5347         -957 | (K,X,Y,R);   77. DIG;  typ: circle
  411- 415| 314    402   5554        -1003 | (K,X,Y,R);   78. DIG;  typ: circle
  416- 420| 300    427   5789        -1159 | (K,X,Y,R);   79. DIG;  typ: circle
  421- 425| 314    498   6009        -1113 | (K,X,Y,R);   80. DIG;  typ: circle
  426- 430| 300    703   6350        -1601 | (K,X,Y,R);   81. DIG;  typ: circle
  431- 435| 314    962   6600        -1307 | (K,X,Y,R);   82. DIG;  typ: circle
  436- 440| 300   1394   6849        -2367 | (K,X,Y,R);   83. DIG;  typ: circle
  441- 445| 314   1901   7017        -2712 | (K,X,Y,R);   84. DIG;  typ: circle
  446- 450| 300   2508   7117        -5432 | (K,X,Y,R);   85. DIG;  typ: circle
  451- 455| 314   3162   7153        -6167 | (K,X,Y,R);   86. DIG;  typ: circle
  456- 460| 300   3679   7129        -5556 | (K,X,Y,R);   87. DIG;  typ: circle
  461- 465| 314   4148   7062        -4662 | (K,X,Y,R);   88. DIG;  typ: circle
  466- 470| 300   4643   6910        -2493 | (K,X,Y,R);   89. DIG;  typ: circle
  471- 475| 314   5037   6684        -1910 | (K,X,Y,R);   90. DIG;  typ: circle
  476- 480| 300   5413   6312        -1883 | (K,X,Y,R);   91. DIG;  typ: circle
  481- 485| 314   5646   5881        -1615 | (K,X,Y,R);   92. DIG;  typ: circle
  486- 490| 300   5758   5465        -2631 | (K,X,Y,R);   93. DIG;  typ: circle
  491- 495| 314   5807   4999        -3124 | (K,X,Y,R);   94. DIG;  typ: circle
  496- 500| 300   5811   4801       -11254 | (K,X,Y,R);   95. DIG;  typ: circle
  501- 505| 315   5813   4526       -21674 | (K,X,Y,R);   96. DIG;  typ: circle
  506- 508| 215   5813   2071              | (K,X,Y);     97. DIG;  typ: vector
  509- 513| 300   5815   1861        14368 | (K,X,Y,R);   98. DIG;  typ: circle
  514- 518| 314   5819   1661        12974 | (K,X,Y,R);   99. DIG;  typ: circle
  519- 523| 300   5836   1406         3341 | (K,X,Y,R);  100. DIG;  typ: circle
  524- 528| 314   5869   1193         2390 | (K,X,Y,R);  101. DIG;  typ: circle
  529- 533| 300   5950    964          855 | (K,X,Y,R);  102. DIG;  typ: circle
  534- 538| 314   6071    807          567 | (K,X,Y,R);  103. DIG;  typ: circle
  539- 543| 300   6218    718          443 | (K,X,Y,R);  104. DIG;  typ: circle
  544- 548| 314   6393    690          476 | (K,X,Y,R);  105. DIG;  typ: circle
  549- 553| 300   6577    717          774 | (K,X,Y,R);  106. DIG;  typ: circle
  554- 558| 314   6747    787          759 | (K,X,Y,R);  107. DIG;  typ: circle
  559- 563| 300   6849    848         2174 | (K,X,Y,R);  108. DIG;  typ: circle
  564- 568| 314   6971    929         3348 | (K,X,Y,R);  109. DIG;  typ: circle
  569- 573| 300   7019    963        16067 | (K,X,Y,R);  110. DIG;  typ: circle
  574- 578| 313   7119   1033        66650 | (K,X,Y,R);  111. DIG;  typ: circle
  579- 581| 112   4513   3884              | (K,X,Y);    112. DIG;  typ: STRP
  582- 584| 215   4513   2715              | (K,X,Y);    113. DIG;  typ: vector
  585- 589| 300   4507   2503        -3665 | (K,X,Y,R);  114. DIG;  typ: circle
  590- 594| 314   4488   2283        -3991 | (K,X,Y,R);  115. DIG;  typ: circle
  595- 599| 300   4428   1930        -3242 | (K,X,Y,R);  116. DIG;  typ: circle
  600- 604| 314   4351   1681        -1713 | (K,X,Y,R);  117. DIG;  typ: circle
  605- 609| 300   4238   1465        -1159 | (K,X,Y,R);  118. DIG;  typ: circle
  610- 614| 314   4056   1237        -1667 | (K,X,Y,R);  119. DIG;  typ: circle
 ---------|-------------------------------|-----------------------------------
```

```
U003003T.DI                   20-JAN-1992  12:00            PAGE  18

 ---------|-------------------------------|-----------------------------------
  WORDNO. |            CONTENT            |              COMMENT
 ---------|-------------------------------|-----------------------------------
  615- 619| 300    3781    1002     -1809 | (K,X,Y,R);  120. DIG;  typ: circle
  620- 624| 314    3462     823     -1852 | (K,X,Y,R);  121. DIG;  typ: circle
  625- 629| 300    3105     709     -1838 | (K,X,Y,R);  122. DIG;  typ: circle
  630- 634| 314    2736     671     -1794 | (K,X,Y,R);  123. DIG;  typ: circle
  635- 639| 300    2328     725     -1571 | (K,X,Y,R);  124. DIG;  typ: circle
  640- 644| 314    1990     872     -1263 | (K,X,Y,R);  125. DIG;  typ: circle
  645- 649| 300    1709    1110     -1278 | (K,X,Y,R);  126. DIG;  typ: circle
  650- 654| 314    1521    1401     -1132 | (K,X,Y,R);  127. DIG;  typ: circle
  655- 659| 300    1423    1707     -1511 | (K,X,Y,R);  128. DIG;  typ: circle
  660- 664| 314    1389    2036     -1604 | (K,X,Y,R);  129. DIG;  typ: circle
  665- 669| 300    1436    2460     -1968 | (K,X,Y,R);  130. DIG;  typ: circle
  670- 674| 314    1555    2782     -1274 | (K,X,Y,R);  131. DIG;  typ: circle
  675- 679| 300    1777    3077     -1185 | (K,X,Y,R);  132. DIG;  typ: circle
  680- 684| 314    2076    3286     -1160 | (K,X,Y,R);  133. DIG;  typ: circle
  685- 689| 300    2368    3408     -2782 | (K,X,Y,R);  134. DIG;  typ: circle
  690- 694| 314    2751    3524     -4463 | (K,X,Y,R);  135. DIG;  typ: circle
  695- 699| 300    3021    3591    -23798 | (K,X,Y,R);  136. DIG;  typ: circle
  700- 704| 314    3226    3639    -13688 | (K,X,Y,R);  137. DIG;  typ: circle
  705- 709| 300    3653    3728    -12820 | (K,X,Y,R);  138. DIG;  typ: circle
  710- 714| 313    4513    3884    -51513 | (K,X,Y,R);  139. DIG;  typ: circle
          |                               |
 ---------|-------------------------------|-----------------------------------
```

```
U003003T.DI                   20-JAN-1992  12:00            PAGE  22

 ---------|------------------|------------------------------------------------
  WORDNO. |     CONTENT      |                     COMMENT
 ---------|------------------|------------------------------------------------
     1    |                 0 | Records      length of character
     2    |               466 | Words
 ---------|------------------|------------------------------------------------
       NAME SECTION
       ------------
     3    |                 3 | Length of name section
     4    |               303 | Character number
     5    |                 0 | Number of following part
 ---------|------------------|------------------------------------------------
       SETTING INFORMATION
       -------------------
     6    |                12 | Length of setting information
     7    |                 1 | Character type = letter
     8    |                89 | Number of DI-points
     9    |              7419 | Total set width (T=L+W+R)
    10    |               680 | Left side bearing (L)
    11    |              6443 | Width (W)
    12    |               296 | Right side bearing (R)
    13    |                 0 | X-minimum
    14    |              6443 | X-maximum
    15    |              -151 | Y-minimum
    16    |              7150 | Y-maximum
    17    |                 1 | Unit is 1/100mm
 ---------|------------------|------------------------------------------------
       CONTOUR INDEX
       -------------
    18    |                 0 | Records     length of contour index
    19    |                 8 | Words
          |                   |
    20    |                 0 | Record pointer for   1. contour:     0
    21    |                 3 | Word pointer                    :    28
    22    |                -1 | Sense of rotation of contour = clockwise
    23    |                 0 | Nesting: outer  1. order
    24    |                 0 | Colour inside: transparent
    25    |                89 | Number of digitizations
          |                   |
 ---------|------------------|------------------------------------------------
       IMAGE INFORMATION
       -----------------
    26    |                              0 | Records lenght of image information
    27    |                            441 | Words
          |                                |
   28-  30| 112    6077   1537             | (K,X,Y);       1. DIG;  typ: STRP
   31-  33| 213    6412   1207             | (K,X,Y);       2. DIG;  typ: vector
   34-  38| 300    6224   1015      -16870 | (K,X,Y,R);     3. DIG;  typ: circle
   39-  43| 314    6077    870       -9980 | (K,X,Y,R);     4. DIG;  typ: circle
   44-  48| 300    5744    583       -3921 | (K,X,Y,R);     5. DIG;  typ: circle
   49-  53| 314    5393    344       -3664 | (K,X,Y,R);     6. DIG;  typ: circle
   54-  58| 300    4965    128       -3351 | (K,X,Y,R);     7. DIG;  typ: circle
   59-  63| 314    4504    -28       -3447 | (K,X,Y,R);     8. DIG;  typ: circle
   64-  68| 300    4028   -118       -3586 | (K,X,Y,R);     9. DIG;  typ: circle
   69-  73| 314    3483   -151       -4558 | (K,X,Y,R);    10. DIG;  typ: circle
   74-  78| 300    2774    -93       -4345 | (K,X,Y,R);    11. DIG;  typ: circle
   79-  83| 314    2116     75       -3966 | (K,X,Y,R);    12. DIG;  typ: circle
   84-  88| 300    1574    330       -2866 | (K,X,Y,R);    13. DIG;  typ: circle
   89-  93| 314    1080    703       -3049 | (K,X,Y,R);    14. DIG;  typ: circle
 ---------|------------------------------|------------------------------------
```

```
U003003T.DI                 20-JAN-1992  12:00            PAGE  24

 ---------|------------------------------|----------------------------------
  WORDNO. |           CONTENT            |            COMMENT
 ---------|------------------------------|----------------------------------
  372- 376| 314   1679   4990       3306 | (K,X,Y,R);   71. DIG;  typ: circle
  377- 381| 300   1545   4279       5801 | (K,X,Y,R);   72. DIG;  typ: circle
  382- 386| 314   1499   3543       6031 | (K,X,Y,R);   73. DIG;  typ: circle
  387- 391| 300   1544   2851       5187 | (K,X,Y,R);   74. DIG;  typ: circle
  392- 396| 314   1676   2214       4564 | (K,X,Y,R);   75. DIG;  typ: circle
  397- 401| 300   1934   1597       2771 | (K,X,Y,R);   76. DIG;  typ: circle
  402- 406| 314   2287   1139       2078 | (K,X,Y,R);   77. DIG;  typ: circle
  407- 411| 300   2620    876       2009 | (K,X,Y,R);   78. DIG;  typ: circle
  412- 416| 314   2983    700       1810 | (K,X,Y,R);   79. DIG;  typ: circle
  417- 421| 300   3435    584       2648 | (K,X,Y,R);   80. DIG;  typ: circle
  422- 426| 314   3936    543       3085 | (K,X,Y,R);   81. DIG;  typ: circle
  427- 431| 300   4368    571       3286 | (K,X,Y,R);   82. DIG;  typ: circle
  432- 436| 314   4721    647       2295 | (K,X,Y,R);   83. DIG;  typ: circle
  437- 441| 300   4993    747       2278 | (K,X,Y,R);   84. DIG;  typ: circle
  442- 446| 314   5286    897       2945 | (K,X,Y,R);   85. DIG;  typ: circle
  447- 451| 300   5548   1068       3182 | (K,X,Y,R);   86. DIG;  typ: circle
  452- 456| 314   5794   1265       3215 | (K,X,Y,R);   87. DIG;  typ: circle
  457- 461| 300   5890   1354       3214 | (K,X,Y,R);   88. DIG;  typ: circle
  462- 466| 313   6077   1537      12884 | (K,X,Y,R);   89. DIG;  typ: circle
          |                              |
 ---------|------------------------------|----------------------------------


                              GRAND TOTAL
                              -----------
                                                  WORDS         KBYTES
                                                  -----         ------
    Length of font:                              299807        599.614
    Length of font header:                         2208          4.416
    Length of all character headers:              21700         43.400
    Length of total image information:           275899        551.798


              DIGITIZATIONS                               NUMBER
              -------------                               ------
              Start points:                                 1362
              Straights:                                    7783
              Arcs:                                        49408
                                                          ======
              Total:                                       58553
              ----------------------------------------------
              Characters:                                    712
              Records (a 2048 words):                        147
              ----------------------------------------------
              Raster (grid):                       15000 X 15000
              ----------------------------------------------
              Digitizations per character:                    82
              Image information per character:               775 bytes



-------------------------------------------------------------------------------
```

Appendix L

Description of the VC format

1. Structure of a VC font

The structure of a font in the **Vector Circle** format is as described in Appendix G, except that the section «font information» in the font header is expanded by the following:

Expanded font information in the font header

Word	Meaning	Comment
	Font information	
I + 2	Length of font information in words	(J)
I + 3	Identification of a font	=1
I + 4	Number of characters	Measured in pixels
I + 5	Cap height	
I + 6	Body size	
I + 7	x-height	
I + 8	Position of base line, as distance from lower body line	
I + 9	Line distance for composition	
I + 10	Stroke width	
I + 11	Optimum size range	in p
I + 12	Italic angle	in 1/10 degree
I + 13	Average character width related to the cap height	in pixels
	New compared with appendix G	
I + 14	X extension of a pixel	in 1/1000 mm
I + 15	Y extension of a pixel	in 1/1000 mm
I + 16	em resolution in X direction	in pixels

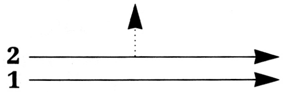
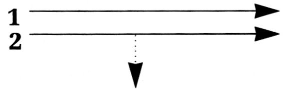
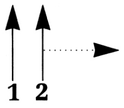
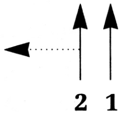
The VC format is a compressed DI format.

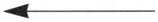
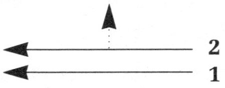
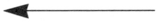
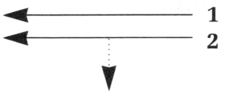
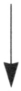
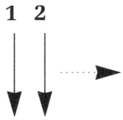
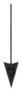
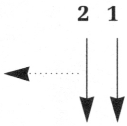
Word	Meaning	Comment
I + 17	em resolution in Y direction	in pixels
I + 18	Scan identification of image data	see below
I + 19	Width of printing window	in pixels
I + 20	Height of printing window	in pixels

The scan identification describes the corner of the em square in which the scanning of the data was actually started, and also the direction in which a scan line will run.

This identification is important for SC, SN, BI and GS formats and is used in those. The following scan identifications are possible:

These scan identifications are possible.

Decimal value	Scanning direction	Direction of line advance
11	→	
12	→	
21	↑	
22	↑	

Decimal value	Scanning direction	Direction of line advance
-11		
-12		
-21		
-22		

Usual values are 21 (CRT machines), 12 (laser printers) and -21 (matrix printers).

The printing window is the frame in which a typesetter can expose. Sometimes laser printers have a smaller printing window than is required for the em of a certain point size. Only then do we describe width and height of the printing window. Otherwise, such details are not necessary.

2. Description of the VC format

The **vector circle format** was created to produce a compressed version of the DI format. The least desirable and most complicated characteristics of the DI format are double words describing radii. Further, it would be advantageous to have a contour format for typesetting machines corresponding to the maximum resolution of the machine right from the start. Such machines do not have a 15,000 × 15,000 unit resolution for an em square but more often 1,000 × 1,000 raster, or even less.

For the above reason, it is appropriate to make three important and basic changes to the DI format: (1) incremental coding, (2) the use of chord heights (instead of radii) describing circle segments and (3) the use of so-called switches. The unit for chord heights is in quarters of the font unit system.

Expansion of font information in the font header also applies to VC, VS, VE, SC, SN, BI and GS formats. Each character has its own coordinate system whose X zero point is at the X minimum of the letter and whose Y zero point is at the baseline.

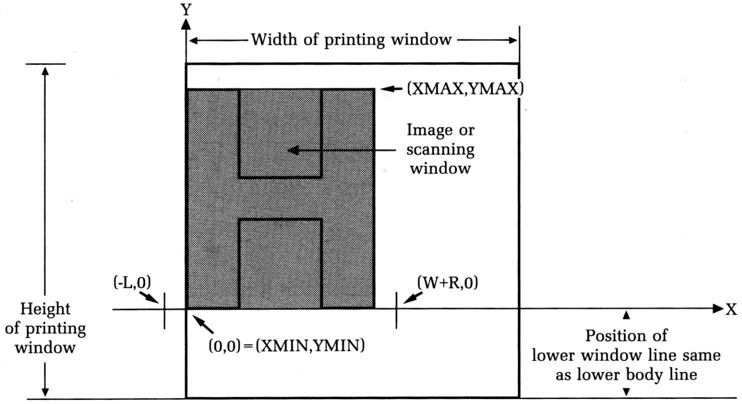

Y
Width of printing window
(XMAX,YMAX)
Image or scanning window
(-L,0)
(W+R,0)
X
Height of printing window
(0,0)=(XMIN,YMIN)
Position of lower window line same as lower body line

Fig. 190
Coordinate system for rasterizing

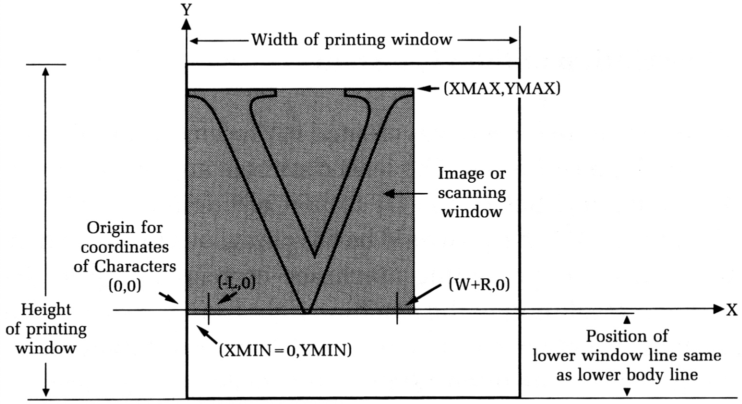

Y
Width of printing window
(XMAX,YMAX)
Image or scanning window
Origin for coordinates of Characters (0,0)
(-L,0)
(W+R,0)
X
Height of printing window
(XMIN=0,YMIN)
Position of lower window line same as lower body line

Fig. 191
Coordinate system for rasterizing

The coding of the straights and circles is done in bytes.

The starting point of contours is reached with «pen raised» incrementally, with a so-called white vector.

The first white vector of a letter starts at the origin (0, 0) of the appropriate coordinate system. The following white vectors start at the ends of preceding contours and lead to following contours in sequence as given in the contour index.

Visible straights are described incrementally as either long or short vectors, circles are described incrementally with a circle vector. We therefore have:

	Switch	ΔX	ΔY	ΔS	Length
W vector		(16 bit)	(16 bit)		= 4 bytes
L vector	127,(7F)	(16 bit)	(16 bit)		= 4(5) bytes
S vector	126,(7E)	(8 bit)	(8 bit)		= 2(3) bytes
C vector	125,(7D)	(8 bit)	(8 bit)	(8 bit)	= 3(4) bytes

Bytes with values 127, 126 and 125 (hexadecimal in brackets) are switches. The switched-on vector applies until switched-off.

An increment which cannot be described in 16 bits is divided into describable 16-bit pieces.

The words (16 bit) and bytes (8 bit) are each given a sign. The words are INTEGER × 2, for the bytes:

values 0...120 are counted as positive and
values 255...136 are counted as negative (-1...-120).

The absolute values of Δ X (8 bit), Δ Y (8 bit) and Δ S (8 bit) have thus to be kept smaller than or equal to 120.

Circles are coded in the form of the above mentioned C vectors. Δ S means the chord height of a related circle segment. If Δ S is positive, the circle turns anti-clockwise (mathematically positive), if Δ S is negative it turns clockwise. The sketch below shows the circle parameters:

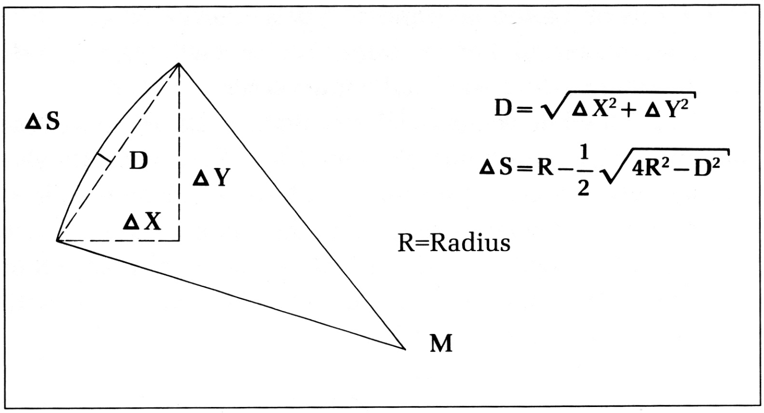

ΔS
D
ΔY
ΔX
M
$D=\sqrt{\Delta X^2+\Delta Y^2}$
$\Delta S=R-\frac{1}{2}\sqrt{4R^2-D^2}$
R=Radius

Fig. 192
Definition of a circle segment by means of chord height

To calculate R (if required) from Δ S a look-up table (LUT) should be used. With it, one can quickly find R as function of Δ S and D.

If a circle (in DI format) has Δ X or Δ Y > 120, then auxiliary points are set along the circle segment such that Δ X and Δ Y ≤ 120.

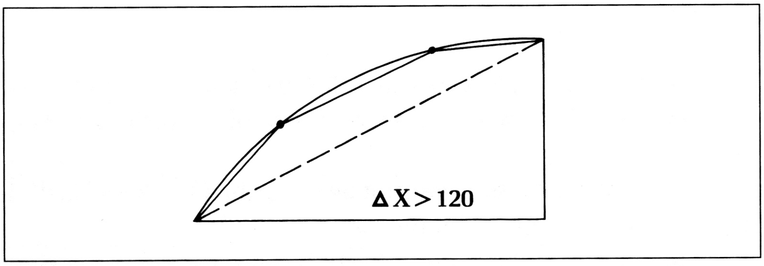


Fig. 193
Vector approximation for circles

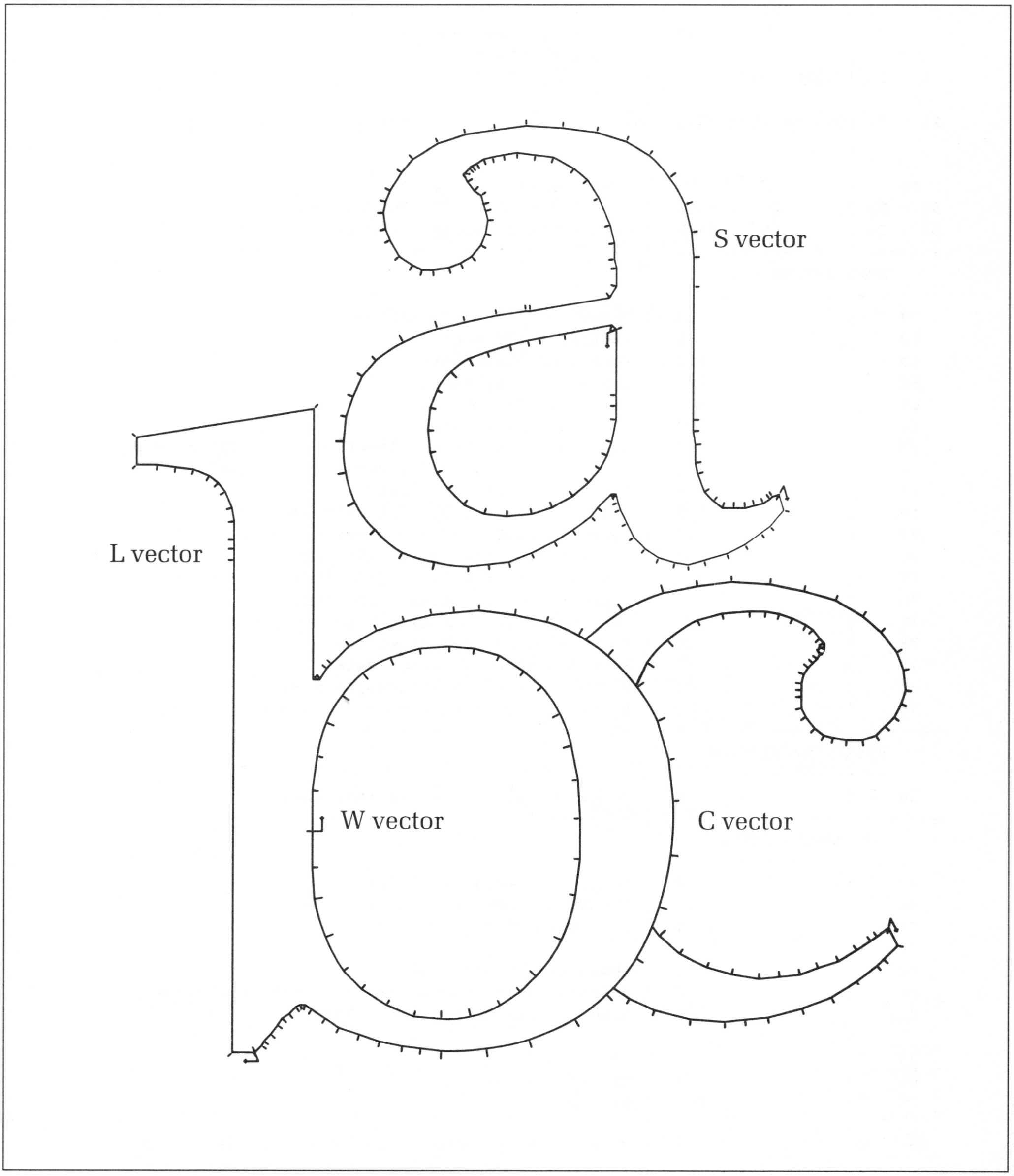
This situation occurs so rarely that we have not included a long C vector for large circles.

At the start of a contour we always have a white vector starting at the word boundary. The end of the contour is controlled by the number of vectors (digitizations); these details are found in the contour index. W, L, S and C vectors are always counted as one digitization. The end of the character is reached when all contours have been processed. The end of a font is checked with the character index of the font.

3. Example of a VC format

Lower case letters «a», «b», «c» of the typeface «URW Antiqua 2015 normal» with marks are shown first, subsequently by the complete font format as a data list. The raster is 400 × 400 for the em.

Fig. 194
Images of
VC format characters


S vector
L vector
W vector
C vector

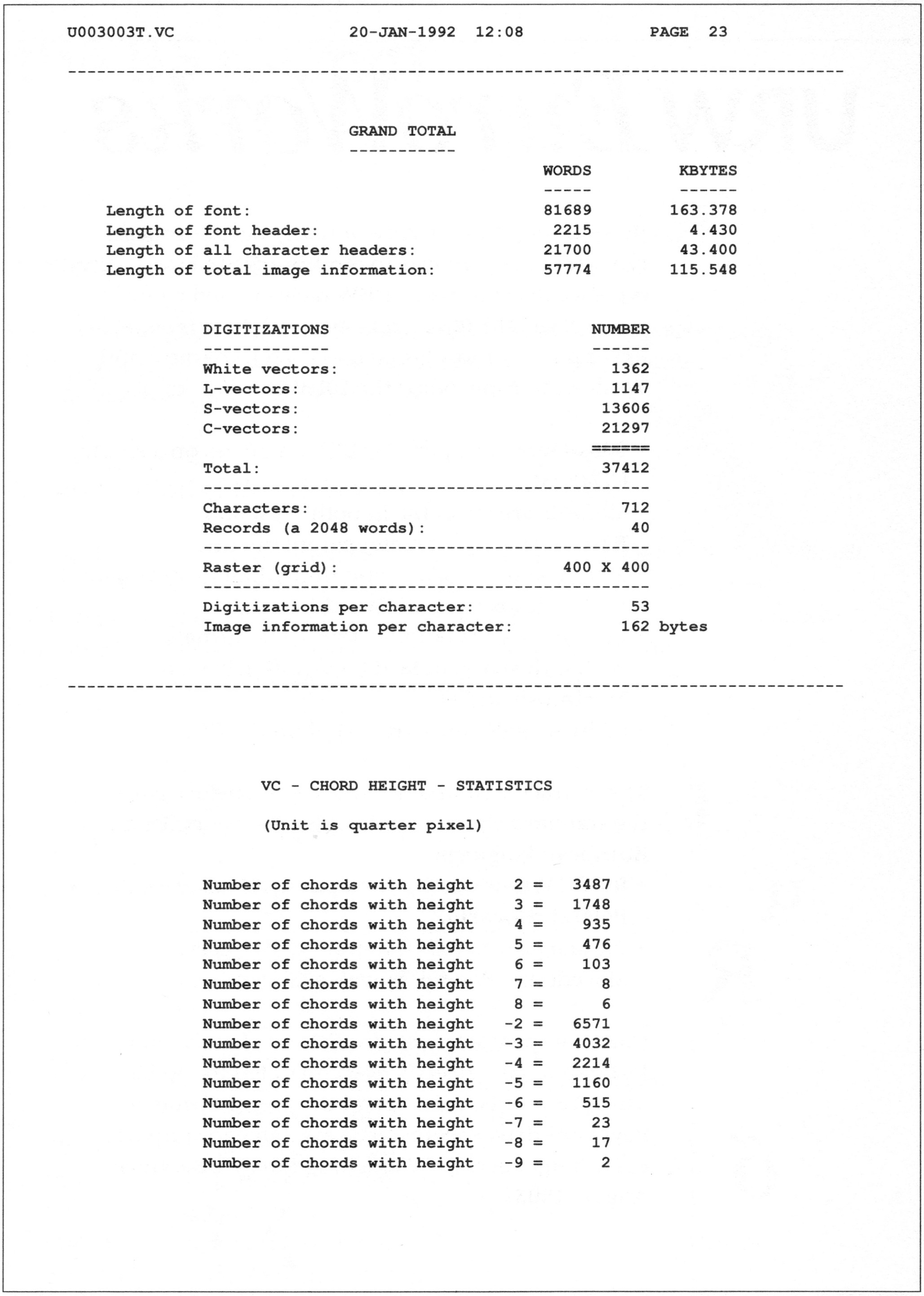
```
U003003T.VC                    20-JAN-1992  12:07            PAGE   1

 ---------|------------------|------------------------------------------------
  WORDNO. |     CONTENT      |                  COMMENT
 ---------|------------------|------------------------------------------------
     1    |             2215 | Length of font header
 ---------|------------------|------------------------------------------------
       NAME SECTION
       ------------
     2    |               55 | Length of name section
     3    |                0 | Number on URW list
   4 - 10 |  U003003T.VC     | File name (ASCII)
          |                  |
  11 - 49 |URW Antiqua 2015 Regular            | Font name (ASCII)
          |
          |                  |
    50    |               VC | Data format (ASCII)
  51 - 53 |       20  1 1992 | Production date:     20-Jan-1992
  54 - 56 |       20  1 1992 | Date of last change: 20-Jan-1992
 ---------|------------------|------------------------------------------------
       FONT INFORMATION
       ----------------
    57    |               19 | Length of font information
    58    |                1 | Indicator for font
    59    |              712 | Number of characters
    60    |              267 | Cap height (in pixels)
    61    |              400 | Body size (in pixels)
    62    |              187 | x-height (in pixels)
    63    |               87 | Distance: base line - lower body line(in pixels)
    64    |              400 | Text line distance for setting (in pixels)
    65    |               35 | Stem thickness (in pixels)
    66    |                0 | Angle of italisation (in 1/10 degree)
    67    |               12 | Optimum point size for setting (in p)
    68    |             8462 | Average character width (in pixels)
    69    |              375 | X-extension of a pixel (in 1/1000mm)
    70    |              375 | Y-extension of a pixel (in 1/1000mm)
    71    |              400 | Em-resolution in X-direction (in pixels)
    72    |              400 | Em-resolution in Y-direction (in pixels)
    73    |               21 | Scan indicator of image data
    74    |              400 | Width of printing window (in pixels)
    75    |              400 | Height of printing window (in pixels)
 ---------|------------------|------------------------------------------------
       HIERARCHY SECTION
       -----------------
    76    |                1 | No hierarchy section in this format
 ---------|------------------|------------------------------------------------
        CHARACTER INDEX
       ----------------
    77    |             2139 | Length of character index
    78    |               40 | LENGTH OF FONT IN PHYSICAL RECORDS
    79    |             1817 | Last word of last record
          |                  |
    80    |              101 | Character number:  1
    81    |                2 | Pointer to record containing start of character
    82    |              168 | Word pointer to start of character
          |                  |
    83    |              102 | Ch. no.: 2
    84    |                2 | Record pointer
    85    |              261 | Word pointer
          |                  |
   86-  88|   103     2  417 | No., record pointer, word pointer of ch.:    3
 ---------|------------------|------------------------------------------------
```

```
U003003T.VC                20-JAN-1992  12:07            PAGE  15

 ---------|-----------------|-----------------------------------------------
  WORDNO. |     CONTENT     |                    COMMENT
 ---------|-----------------|-----------------------------------------------
     1    |              0  | Records      length of character
     2    |            160  | Words
 ---------|-----------------|-----------------------------------------------
       NAME SECTION
       ------------
     3    |              3  | Length of name section
     4    |            301  | Character number
     5    |              0  | Number of following part
 ---------|-----------------|-----------------------------------------------
       SETTING INFORMATION
       -------------------
     6    |             12  | Length of setting information
     7    |              1  | Character type = letter
     8    |             84  | Number of vectors
     9    |            220  | Total set width (T=L+W+R)
    10    |             21  | Left side bearing (L)
    11    |            194  | Width (W)
    12    |              5  | Right side bearing (R)
    13    |              0  | X-minimum
    14    |            194  | X-maximum
    15    |             -4  | Y-minimum
    16    |            191  | Y-maximum
    17    |              1  | Unit is 1 pixel
 ---------|-----------------|-----------------------------------------------
       CONTOUR INDEX
       -------------
    18    |              0  | Records     length of contour index
    19    |             14  | Words
          |                 |
    20    |              0  | Record pointer for   1. contour:     0
    21    |              3  | Word pointer                   :    34
    22    |             -1  | Sense of rotation of contour = clockwise
    23    |              0  | Nesting: outer  1. order
    24    |              0  | Colour inside: transparent
    25    |             69  | Number of digitizations
          |                 |
    26    |              0  | Record pointer for   2. contour:     0
    27    |            105  | Word pointer                   :   136
    28    |             -1  | Sense of rotation of contour = clockwise
    29    |              1  | Nesting: inner  1. order
    30    |              0  | Colour inside: transparent
    31    |             15  | Number of digitizations
          |                 |
 ---------|-----------------|-----------------------------------------------
       IMAGE INFORMATION
       -----------------
    32    |              0  | Records     length of image information
    33    |            129  | Words
          |                 |
   34- 135|      190     28 |        190    28   (dX,dY)       1.DIG w-vector
    36 1  | 7E  04   F9     | 126   4  -7     (K,dX,dY)       2.DIG s-vector
    37 2  | 7D  F0   F2  FE | 125 -16 -14  -2 (K,dX,dY,dS)    3.DIG c-vector
    39 2  |     ED   F7  FD |     -19  -9  -3   (dX,dY,dS)    4.DIG c-vector
    41 1  | 7E  F7   FF     | 126  -9  -1     (K,dX,dY)       5.DIG s-vector
    42 2  | 7D  F5   03  FD | 125 -11   3  -3 (K,dX,dY,dS)    6.DIG c-vector
    44 2  |     F4   0A  FC |     -12  10  -4   (dX,dY,dS)    7.DIG c-vector
 ---------|-----------------|-----------------------------------------------
```

```
U003003T.VC                    20-JAN-1992  12:07            PAGE  16

 ---------|------------------|-------------------------------------------------
 WORDNO.  |      CONTENT     |                    COMMENT
 ---------|------------------|-------------------------------------------------
    46 1  |      FB   0B  FE |        -5  11  -2   (dX,dY,dS)       8.DIG c-vector
    47 2  | 7E   FF   04     | 126    -1   4      (K,dX,dY)        9.DIG s-vector
    49 1  |      FF   05     |        -1   5       (dX,dY)        10.DIG s-vector
    50 1  | 7D   EE   EF  FE | 125   -18 -17  -2 (K,dX,dY,dS)     11.DIG c-vector
    52 1  |      ED   F5  FD |       -19 -11  -3   (dX,dY,dS)      12.DIG c-vector
    53 2  |      ED   FB  FD |       -19  -5  -3   (dX,dY,dS)      13.DIG c-vector
    55 1  | 7E   F6   FF     | 126   -10  -1      (K,dX,dY)       14.DIG s-vector
    56 2  | 7D   F1   02  FE | 125   -15   2  -2 (K,dX,dY,dS)     15.DIG c-vector
    58 2  |      F4   04  FE |       -12   4  -2   (dX,dY,dS)      16.DIG c-vector
    60 1  |      F3   09  FD |       -13   9  -3   (dX,dY,dS)      17.DIG c-vector
    61 2  |      F7   0E  FD |        -9  14  -3   (dX,dY,dS)      18.DIG c-vector
    63 1  |      FB   19  FB |        -5  25  -5   (dX,dY,dS)      19.DIG c-vector
    64 2  |      04   16  FC |         4  22  -4   (dX,dY,dS)      20.DIG c-vector
    66 1  |      0A   0F  FB |        10  15  -5   (dX,dY,dS)      21.DIG c-vector
    67 2  |      0C   08  FE |        12   8  -2   (dX,dY,dS)      22.DIG c-vector
    69 1  |      0F   06  FE |        15   6  -2   (dX,dY,dS)      23.DIG c-vector
    70 2  | 7E   0C   03     | 126    12   3      (K,dX,dY)       24.DIG s-vector
    72 1  |      0F   03     |        15   3       (dX,dY)        25.DIG s-vector
    73 1  |      08   01     |         8   1       (dX,dY)        26.DIG s-vector
    74 1  |      06   01     |         6   1       (dX,dY)        27.DIG s-vector
    75 1  |      26   07     |        38   7       (dX,dY)        28.DIG s-vector
    76 1  | 7D   FE   20  02 | 125    -2  32   2 (K,dX,dY,dS)     29.DIG c-vector
    78 1  |      FC   0C  02 |        -4  12   2   (dX,dY,dS)      30.DIG c-vector
    79 2  |      FA   09  02 |        -6   9   2   (dX,dY,dS)      31.DIG c-vector
    81 1  |      F0   08  05 |       -16   8   5   (dX,dY,dS)      32.DIG c-vector
    82 2  |      F1   02  02 |       -15   2   2   (dX,dY,dS)      33.DIG c-vector
    84 1  |      ED   FD  03 |       -19  -3   3   (dX,dY,dS)      34.DIG c-vector
    85 2  |      F7   FB  02 |        -9  -5   2   (dX,dY,dS)      35.DIG c-vector
    87 1  |      FD   FC  02 |        -3  -4   2   (dX,dY,dS)      36.DIG c-vector
    88 2  | 7E   00   FE     | 126     0  -2      (K,dX,dY)       37.DIG s-vector
    90 1  |      02   FE     |         2  -2       (dX,dY)        38.DIG s-vector
    91 1  |      03   FD     |         3  -3       (dX,dY)        39.DIG s-vector
    92 1  | 7D   04   FA  FE | 125     4  -6  -2 (K,dX,dY,dS)     40.DIG c-vector
    94 1  | 7E   01   FC     | 126     1  -4      (K,dX,dY)       41.DIG s-vector
    95 2  |      00   FC     |         0  -4       (dX,dY)        42.DIG s-vector
    96 2  | 7D   FE   F7  FD | 125    -2  -9  -3 (K,dX,dY,dS)     43.DIG c-vector
    98 2  |      F6   F6  FC |       -10 -10  -4   (dX,dY,dS)      44.DIG c-vector
   100 1  |      F4   FE  FD |       -12  -2  -3   (dX,dY,dS)      45.DIG c-vector
   101 2  |      F7   02  FE |        -9   2  -2   (dX,dY,dS)      46.DIG c-vector
   103 1  |      F6   0A  FB |       -10  10  -5   (dX,dY,dS)      47.DIG c-vector
   104 2  |      FE   0C  FE |        -2  12  -2   (dX,dY,dS)      48.DIG c-vector
   106 1  |      02   0B  FD |         2  11  -3   (dX,dY,dS)      49.DIG c-vector
   107 2  |      0D   10  FB |        13  16  -5   (dX,dY,dS)      50.DIG c-vector
   109 1  |      18   0B  FA |        24  11  -6   (dX,dY,dS)      51.DIG c-vector
   110 2  |      22   04  FC |        34   4  -4   (dX,dY,dS)      52.DIG c-vector
   112 1  |      1B   FD  FD |        27  -3  -3   (dX,dY,dS)      53.DIG c-vector
   113 2  |      18   F6  FA |        24 -10  -6   (dX,dY,dS)      54.DIG c-vector
   115 1  |      09   F7  FE |         9  -9  -2   (dX,dY,dS)      55.DIG c-vector
   116 2  |      07   F3  FE |         7 -13  -2   (dX,dY,dS)      56.DIG c-vector
   118 1  |      04   E9  FC |         4 -23  -4   (dX,dY,dS)      57.DIG c-vector
   119 2  | 7E   00   FB     | 126     0  -5      (K,dX,dY)       58.DIG s-vector
   121 1  |      00   F8     |         0  -8       (dX,dY)        59.DIG s-vector
   122 1  |      00   C0     |         0 -64       (dX,dY)        60.DIG s-vector
   123 1  |      00   FA     |         0  -6       (dX,dY)        61.DIG s-vector
   124 1  | 7D   02   ED  02 | 125     2 -19   2 (K,dX,dY,dS)     62.DIG c-vector
   126 1  |      04   F7  04 |         4  -9   4   (dX,dY,dS)      63.DIG c-vector
 ---------|------------------|-------------------------------------------------
```

```
U003003T.VC                  20-JAN-1992  12:07            PAGE  17

---------|------------------|------------------------------------------------
 WORDNO. |       CONTENT    |                  COMMENT
---------|------------------|------------------------------------------------
   127 2 |      09   FC  04 |         9  -4   4   (dX,dY,dS)     64.DIG c-vector
   129 1 |      0A   03  02 |        10   3   2   (dX,dY,dS)     65.DIG c-vector
   130 2 | 7E   03   02     | 126     3   2     (K,dX,dY)        66.DIG s-vector
   132 1 |      02   01     |         2   1       (dX,dY)        67.DIG s-vector
   133 1 |      03   02     |         3   2       (dX,dY)        68.DIG s-vector
   134 1 |      02   02     |         2   2       (dX,dY)        69.DIG s-vector
 136- 160|       -70      76 |       -70      76   (dX,dY)        70.DIG w-vector
   138 1 | 7E   00   E1     | 126     0 -31     (K,dX,dY)        71.DIG s-vector
   139 2 | 7D   FE   EA  FE | 125    -2 -22  -2 (K,dX,dY,dS)     72.DIG c-vector
   141 2 |      FB   F4  FE |        -5 -12  -2   (dX,dY,dS)     73.DIG c-vector
   143 1 |      F3   F3  FD |       -13 -13  -3   (dX,dY,dS)     74.DIG c-vector
   144 2 |      EF   F9  FC |       -17  -7  -4   (dX,dY,dS)     75.DIG c-vector
   146 1 | 7E   F5   FF     | 126   -11  -1     (K,dX,dY)        76.DIG s-vector
   147 2 | 7D   ED   05  FA | 125   -19   5  -6 (K,dX,dY,dS)     77.DIG c-vector
   149 2 |      F3   0F  FA |       -13  15  -6   (dX,dY,dS)     78.DIG c-vector
   151 1 |      FD   10  FC |        -3  16  -4   (dX,dY,dS)     79.DIG c-vector
   152 2 |      05   15  FB |         5  21  -5   (dX,dY,dS)     80.DIG c-vector
   154 1 |      05   07  FE |         5   7  -2   (dX,dY,dS)     81.DIG c-vector
   155 2 |      09   06  FE |         9   6  -2   (dX,dY,dS)     82.DIG c-vector
   157 1 |      11   06  FE |        17   6  -2   (dX,dY,dS)     83.DIG c-vector
   158 2 |      30   0A  FE |        48  10  -2   (dX,dY,dS)     84.DIG c-vector
         |                  |
---------|------------------|------------------------------------------------
```

```
     U003003T.VC                   20-JAN-1992  12:07             PAGE  18

  ---------|------------------|------------------------------------------------
   WORDNO. |      CONTENT     |                     COMMENT
  ---------|------------------|------------------------------------------------
      1    |                 0| Records      length of character
      2    |               132| Words
  ---------|------------------|------------------------------------------------
        NAME SECTION
        ------------
      3    |                 3| Length of name section
      4    |               302| Character number
      5    |                 0| Number of following part
  ---------|------------------|------------------------------------------------
        SETTING INFORMATION
        -------------------
      6    |                12| Length of setting information
      7    |                 1| Character type = letter
      8    |                66| Number of vectors
      9    |               251| Total set width (T=L+W+R)
     10    |                -1| Left side bearing (L)
     11    |               234| Width (W)
     12    |                18| Right side bearing (R)
     13    |                 0| X-minimum
     14    |               234| X-maximum
     15    |                -5| Y-minimum
     16    |               276| Y-maximum
     17    |                 1| Unit is 1 pixel
  ---------|------------------|------------------------------------------------
        CONTOUR INDEX
        -------------
     18    |                 0| Records     length of contour index
     19    |                14| Words
           |                  |
     20    |                 0| Record pointer for   1. contour:     0
     21    |                 3| Word pointer                   :    34
     22    |                -1| Sense of rotation of contour = clockwise
     23    |                 0| Nesting: outer  1. order
     24    |                 0| Colour inside: transparent
     25    |                42| Number of digitizations
           |                  |
     26    |                 0| Record pointer for   2. contour:     0
     27    |                64| Word pointer                   :    95
     28    |                -1| Sense of rotation of contour = clockwise
     29    |                 1| Nesting: inner  1. order
     30    |                 0| Colour inside: transparent
     31    |                24| Number of digitizations
           |                  |
  ---------|------------------|------------------------------------------------
        IMAGE INFORMATION
        -----------------
     32    |                 0| Records     length of image information
     33    |               101| Words
           |                  |
    34-  94|      51      -5  |          51    -5   (dX,dY)        1.DIG w-vector
     36 1  | 7E   F6   00     | 126  -10    0      (K,dX,dY)       2.DIG s-vector
     37 2  | 7F     0    219  | 127     0   219 (K,dX,dY)       3.DIG l-vector
     40 1  | 7E   00   04     | 126     0    4      (K,dX,dY)       4.DIG s-vector
     41 2  | 7D   FE   13   02| 125    -2   19    2 (K,dX,dY,dS)    5.DIG c-vector
     43 2  |      FB   0A   04|        -5   10    4   (dX,dY,dS)    6.DIG c-vector
     45 1  |      F4   06   03|       -12    6    3   (dX,dY,dS)    7.DIG c-vector
  ---------|------------------|------------------------------------------------
```



```
U003003T.VC                20-JAN-1992  12:08            PAGE  23

-----------------------------------------------------------------------------


                           GRAND TOTAL
                           -----------
                                                 WORDS          KBYTES
                                                 -----          ------
    Length of font:                              81689         163.378
    Length of font header:                        2215           4.430
    Length of all character headers:             21700          43.400
    Length of total image information:           57774         115.548


              DIGITIZATIONS                          NUMBER
              -------------                          ------
              White vectors:                           1362
              L-vectors:                               1147
              S-vectors:                              13606
              C-vectors:                              21297
                                                     ======
              Total:                                  37412
              ---------------------------------------------
              Characters:                               712
              Records (a 2048 words):                    40
              ---------------------------------------------
              Raster (grid):                      400 X 400
              ---------------------------------------------
              Digitizations per character:               53
              Image information per character:          162 bytes


-----------------------------------------------------------------------------



                    VC - CHORD HEIGHT - STATISTICS

                    (Unit is quarter pixel)


              Number of chords with height    2 =   3487
              Number of chords with height    3 =   1748
              Number of chords with height    4 =    935
              Number of chords with height    5 =    476
              Number of chords with height    6 =    103
              Number of chords with height    7 =      8
              Number of chords with height    8 =      6
              Number of chords with height   -2 =   6571
              Number of chords with height   -3 =   4032
              Number of chords with height   -4 =   2214
              Number of chords with height   -5 =   1160
              Number of chords with height   -6 =    515
              Number of chords with height   -7 =     23
              Number of chords with height   -8 =     17
              Number of chords with height   -9 =      2
```

URW *EuroWorks*

By applying the expertise of many East European typographic specialists to our library of high-quality typeface designs, we at URW have created a new package of DTP fonts and keyboard drivers capable of supporting most languages used in Eastern and Western Europe today: the URW EuroWorks CD.

- 500 of our most popular DTP typefaces on a single CD-ROM,
- all fonts are available in both West and East European character complements,
- all fonts are unlocked and immediately accessible,
- for Macintosh and PC Windows,
- in PostScript Type1 and TrueType formats,
- with industry standard keyboard drivers for Macintosh and PC,
- all in all 4000 fonts are stored on the CD.

The character complement of each typeface covers the standard character sets of the following East European languages:
- for PC Windows: Czech Republic, Hungary, Poland, Slovakia.
- for Macintosh: Czech Republic, Hungary, Macedonia, Poland, Slovakia, Slovenia.

Complete character sets for Estonia, Latvia, and Lithuania have already been included in the East European fonts, although at the present time no keyboard driver seems to be available to support these languages. URW is selling EuroWorks since August 1993.

Description of the BS format

1. Structure of a BS format

The structure of the **Bézier Scaling (BS) format** has a basis similar to the VC format, it does contain many special features and is therefore described here in detail.

2. Description of the BS Format

The BS format contains instructions.

A font is stored as a direct-access data set and has a record length of 512 bytes. Most data pointers are 16-bit integer values and point to bytes. The first byte in a data set has address zero (0). The font data set is divided into three parts, each beginning at an integral boundary (one word = two bytes). The three parts are the:

- Font header,
- Character index, and
- Character data.

Font Header

Byte		Meaning
0 - 3		Length of the font in bytes
4 - 5		System code (0 = Mac, 257 = VAX)
6 - 7	lt =	Length of the font header (currently ≥ 152)
8 - 19		File name (12 bytes) [ASCII]
20 - 97		Font name [ASCII]
98 - 99		Format extension [ASCII]
100 - 105		Production date
106 - 111		Date of last change
112 - 113		Number of characters

Byte	Meaning
114 - 115	Body size
116 - 117	Cap height
118 - 119	x-height
120 - 121	Distance baseline - lower body (negative, the baseline is at 0)
122 - 123	Italic angle
124 - 125	Em resolution in X direction per character
126 - 127	Em resolution in Y direction
128 - 129	X-minimum of font bounding box
130 - 131	X-maximum of font bounding box
132 - 133	Y-minimum of font bounding box
134 - 135	Y-maximum of font bounding box
136 - 137	Maximum number of serifs per character
138 - 139	Maximum number of peaks per character
140 - 141	Maximum number of curves per character
142 - 143	Maximum number of horizontal curve extrema
144 - 145	Maximum number of vertical curve extrema per character
146 - 147	Maximum number of contours per character
148 - 149	Number of additional, invisible horizontal lines
150 - 151	Position of the first invisible horizontal line
... - ...	Total number of invisible vertical lines
... - ...	Position of the first invisible vertical line.
	Space for extensions

Character Index

Byte	Meaning
lt + 0,1,2,3	ld = Length of character index
lt + 4,5	Number of first character
lt + 6,7,8,9	Pointer to image information of first character

Byte	Meaning
lt + 10,11	Length of image information of first character
lt + 12,13	Number of second character
lt + 14,15,16,17	Pointer to second character
lt + 18,19	Length of second character
lt + 20,21	Number of third character
.	.
etc.	etc.

Character numbers are unsigned two-byte words with values between 1 and 65,535.

Character Data

The character data is divided into four sections. Starting at an integral boundary, each section comprises a:

- Character header,
- Stem block,
- Contour index, and
- Image information.

Character Header

Character data starts at the byte position defined in the character index. The following byte addresses are all relative to this position. The character header has a fixed length of 14 bytes.

Byte	Meaning
0, 1	Control word Bit 0 set: upper case character Bit 1 set: lower case character Bits 2–15: not used
2, 3	Number of digitizations
4, 5	Total width
6, 7	Left side bearing

Byte	Meaning
8, 9	Width
10, 11	Y-minimum
12, 13	Y-maximum
	lb = 14, fixed length of character header

Note: All entries in the character header are two-byte words. X-minimum is, by definition, always 0; X-maximum corresponds to the width.

Stem Block

The stem block may consist of several sections, each corresponding to a group of contours. The groups are formed in such a way that all the inner contour groups pertaining to an outer contour follow on from the outer contour to which they belong. If the first group consists of two contours, they are denoted by 1, 2. If the next group in a character consists of three contours, these are assigned 3, 4, 5.

Byte	Meaning
lb + 0, 1	ls = Length of the stem block
lb + 2	Number of contours in the first group (1 byte)
+ 3	Number of visible vertical straights (1 byte)
+ 4, 5	Index of the first visible vertical straight (2 bytes)
+ 6, 7	Index of the second visible vertical straight (where present, 2 bytes)
...	...
...	Number of visible horizontal straights (1 byte)
...	Index of the first horizontal straight (2 bytes)
...	Index of the second horizontal straight (where present, 2 bytes)

Further sections follow if other contour groups occur and the current byte address is smaller than «ls».

Notes: In a character not including any straights, the number of contours is coded negatively, and no further data follows. The number of straights is always coded with one byte, the indices of straights with two bytes. The stem block contains visible lines only. Invisible lines, such as the baseline, ascender, descender, x-height and cap height lines, are defined in the font header.

The stem edges, for instance, are visible straights. Their indices are coded with two bytes as shown below:

Bit 15	0 for absolute position 1 for relative position
Bits 14, 13	Color of corresponding stem 00 = gray 10 = left or below is black 01 = left or below is white
Bits 12,...,0	Address of digitization which defines this straight (this explains why the number of digitizations is restricted to 8192)

Relative straights (bit 15 = 1) then have a further index defining the corresponding absolute straight. This index is coded with either one or two bytes, depending on how many digitizations and invisible lines there are. It is also possible to define relations to invisible lines (see example below for image information coding).

Contour Index

Byte	Comment
lb + ls + 0, 1	lc = Length of the contour index

Normal contours have a normal contour index in the image information, but no entry here in the stem block. Only if we have a pointer to another element, accent or character is a contour index specified here (as in the II format).

The contour index consists of 12 bytes with the following meaning:

0, 1	Character number
2, 3	X-position of the lower left corner of bounding box
4, 5	Y-position of the lower left corner of bounding box
6, 7	X-factor
8, 9	Y-factor
10, 11	Angle of rotation (or mirroring code)
	The X and Y factors are fixed-point numbers:
Bits 12 - 15	form the integer number (before the decimal point)
Bits 0 - 11	form the fraction (after the decimal point)

The angle of rotation is coded in units of 1/50 degrees and is greater than zero. Negative values for the angle of rotation denote coding for mirroring functions:

-1	Mirror left/right
-2	Mirror top/bottom
-3	Both directions simultaneously.

Mirroring is implemented within the bounding box of this element. The direction of rotation of a contour is always such that black is on the right (in the direction of movement).

Image Information

Byte	Comment
lb + ls + lc + 0, 1	Length of image information

Each digitization is generally coded as follows:

1st byte	Control byte
2nd byte	X-component
3rd byte	Y-component
4th byte	Instruction for «intelligent scaling»
5th byte	Index of corresponding point.

Each coordinate is coded relative to the preceding one. X and Y components do not apply when they equal zero. Otherwise they are always unsigned integers (signs are given in the control byte). Bytes 4 and 5 are used only when indicated in the control byte. The start position of a contour is given only once. As all contours are closed, the final position in the contour must be linked with the start position.

The contour is described using straights and Bézier curves. In BS format, a Bézier curve is described using two anchor points (points on the contour) and two control points (off-contour). The control byte determines whether a point is a Bézier control point (bit 7 = 1).

The starting point of the contours is described with a so-called white vector (pen raised). The first white vector is absolute, but all following vectors are coded relatively, i.e. they lead from the start point of one contour to the start point of the next.

The first byte is the control byte:

Bit 0 = 1	X = 0
Bit 1 = 1	Y = 0
Bit 0 = 1 and Bit 2 = 1	The following bytes form a character pointer.
Bit 2 = 1	X coded with 2 bytes
Bit 3 = 1	Y coded with 2 bytes
Bit 4 = 1	X negative
Bit 5 = 1	Y negative
Bit 6 = 1	Instructed digitization
Bit 7 = 1	Digitization is off-contour.

Each white vector has two control bytes, the first being 0xFF.

Future extensions can be described with the following two bits (combined):

Bit 6 + 7, bit 1 + 3, bit 0 + 4, bit 1 + 5.

The instruction byte contains the following:

Bit 7 = 1	Further instruction byte follows
Bit 6 = 1	Curve extremum
Bit 5 = 1	Serif
Bit 4 = 1	Peak
Bit 3 = 1	This digitization must be scaled relative to another.
Bit 2 = 1	Relative scaling in X-direction, otherwise in Y

Bit 1, 0	Color information: 00 = gray 10 = left or below is black 01 = left or below is white.
Bit 5 = 1 and Bit 2 = 1	End of a serif,
Bit 5 = 1 and Bit 2 = 0	Start of a serif.

Bit 4 = 1 (peak) is set to indicate that a digitization point forms both the end of one curve and the start of the next. When bit 3 = 1 is set, the index of the corresponding digitization follows on immediately as a one or two-byte value.

The number of digitizations and the number of invisible lines (in the font header) determine whether a one or two byte value is used. These indices are only coded with two bytes when the number of digitizations plus the number of invisible horizontal or vertical lines is greater than 255.

Two bytes	One byte	Comment
65535	255	Index for baseline or 1st invisible vertical line
65534	254	Index for cap height or 2nd invisible vertical line
65533	253	Index for x-height or 3rd invisible vertical line
65532	252	Index for descender or 4th invisible vertical line
65531	251	Index for first additional invisible horizontal line or 5th invisible vertical line
65532 – n	252 – n	Index for n-th additional invisible horizontal line or (n + 4)th invisible vertical line

All other indices refer to given digitization points.

Comment: All digitization points denoting a peak, the Y-anchor point of a serif or a curve extremum need information as to their color, as this enables NIMBUS to rasterize the characters more effectively:

Bit 2 =1	With peaks and curve extrema, the color information applies to the left side of the digitization.
Bit 2 = 0	With peaks, curve extrema and serif anchor points, the color information applies to the lower side of the digitization.

If the entire contour is round (no straight lines or corners), the white vector points to a curve point. Note that:

- White vectors may not point to a serif,
- White vectors may not point to curve points other than curve extrema,
- Digitizations may not be located within a serif when they are of the peak or curve extremum point types.
- Digitizations may not be located within a curve when they are of the peak or serif anchor point types.

Only one of the bits 4, 5 and 6 may be set for each digitization.

When a white vector is assigned the serif point type, NIMBUS will interpret this instruction as applying to the end point of the associated contour. A serif thus ends at the start point of a contour when the start point has been marked as a serif point. The first serif does not actually commence until the next serif instruction following the white vector.

3. Example of a BS Format

The lower case characters «a», «b» and «c» of the «URW Antiqua 2015 (normal)» font as contour diagrams with symbols for the instructions are shown first. We then give the complete font format as a data list. The em resolution is 1024 × 1024.

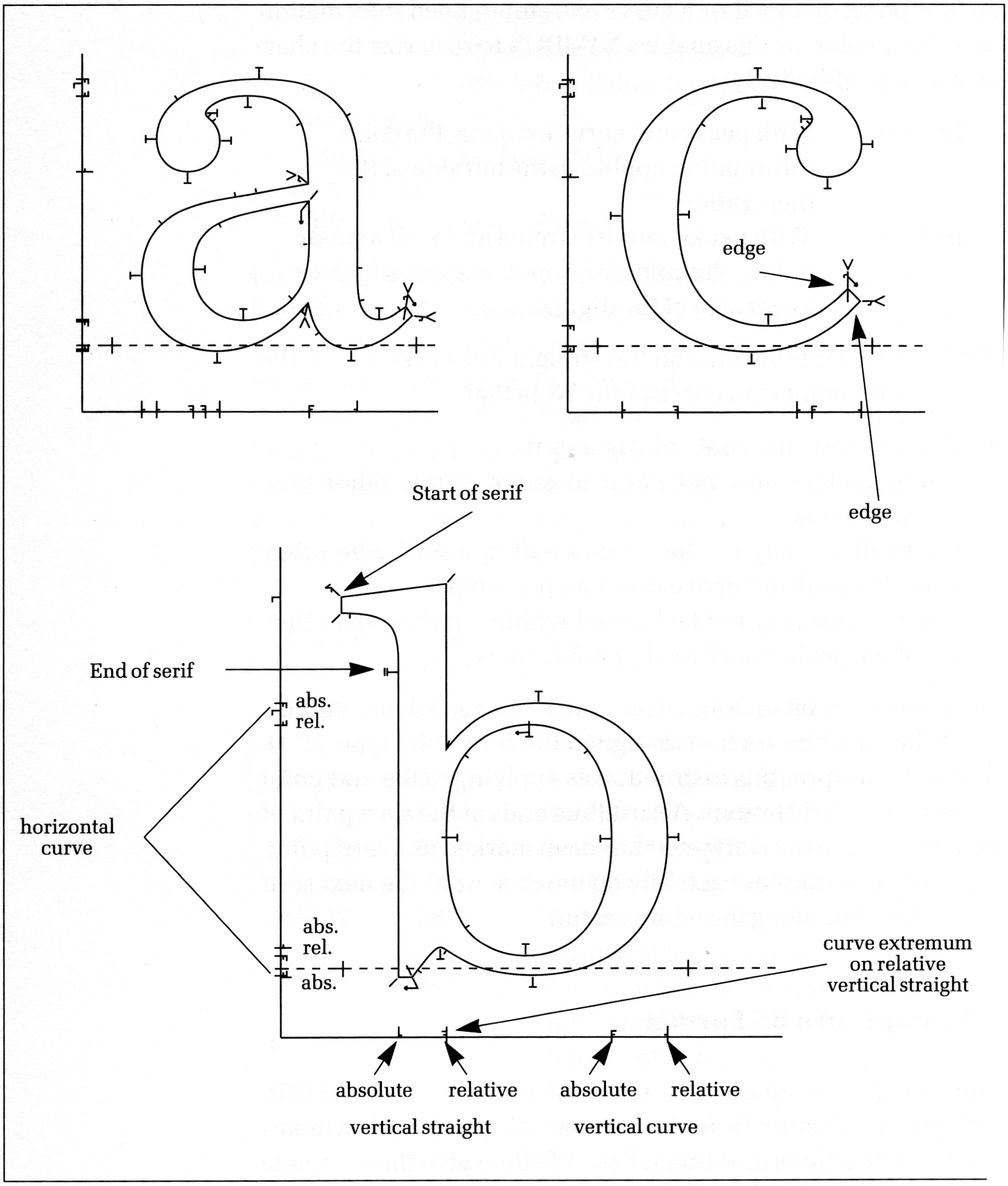
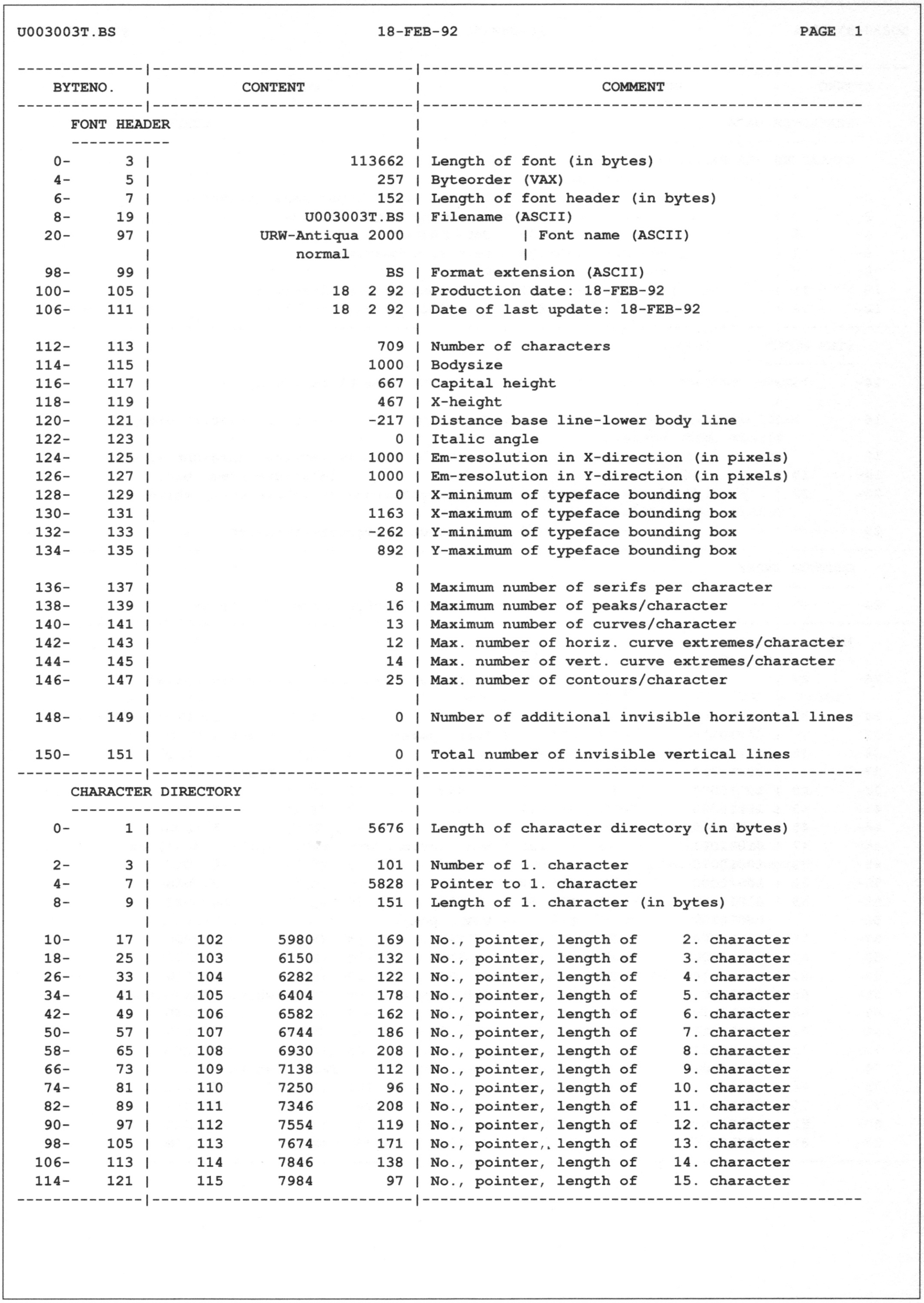
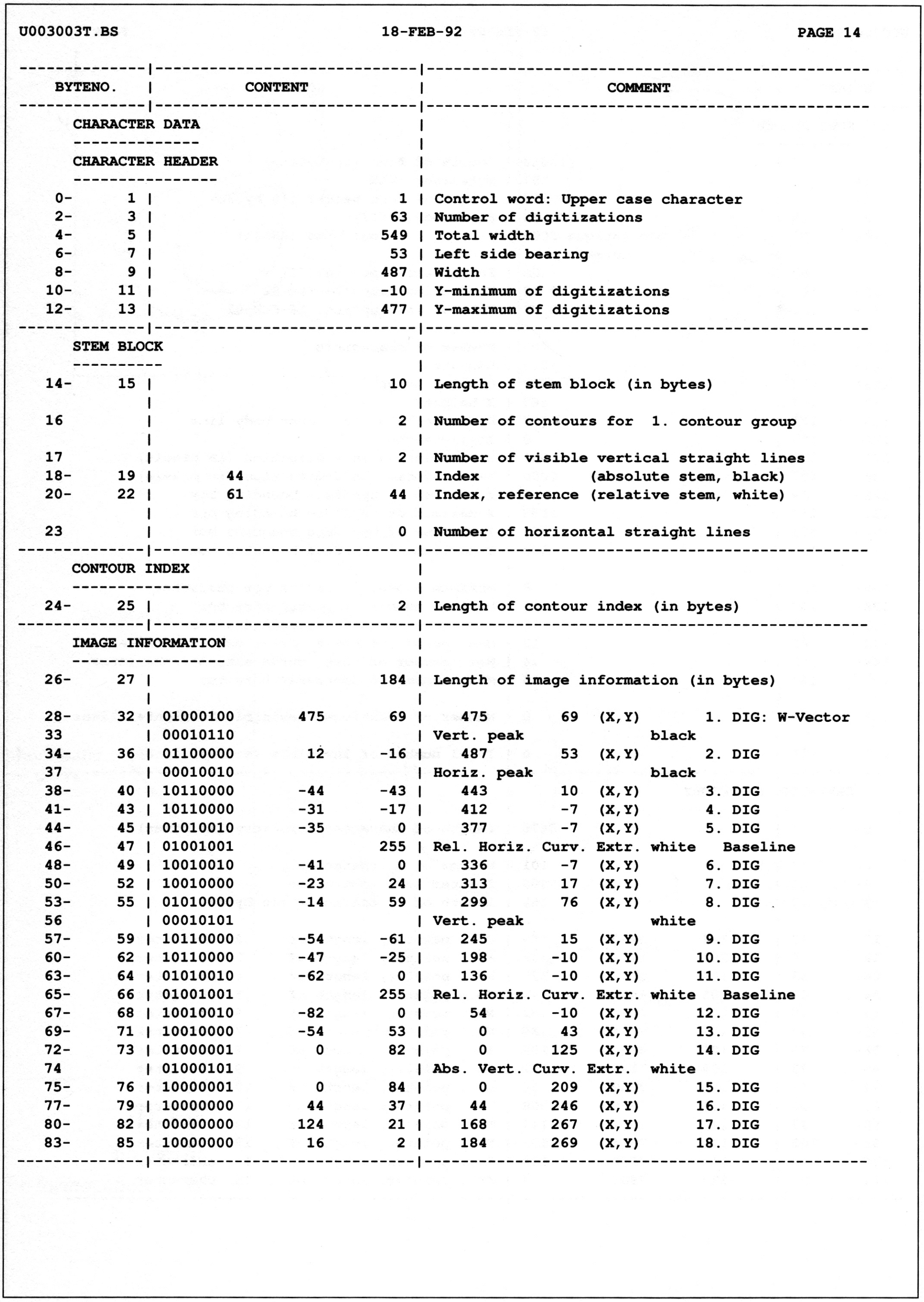

edge
Start of serif
edge
End of serif
abs.
rel.
horizontal curve
abs.
rel.
abs.
curve extremum on relative vertical straight
absolute
relative
absolute
relative
vertical straight
vertical curve

Abb. 195
Images of
BS format characters



```
U003003T.BS                            18-FEB-92                              PAGE  1

--------------|----------------------------|------------------------------------------------
   BYTENO.    |          CONTENT           |                   COMMENT
--------------|----------------------------|------------------------------------------------
      FONT HEADER                          |
      -----------                          |
    0-     3 |                      113662 | Length of font (in bytes)
    4-     5 |                         257 | Byteorder (VAX)
    6-     7 |                         152 | Length of font header (in bytes)
    8-    19 |                 U003003T.BS | Filename (ASCII)
   20-    97 |            URW-Antiqua 2000             | Font name (ASCII)
             |                normal       |
   98-    99 |                          BS | Format extension (ASCII)
  100-   105 |                     18  2 92 | Production date: 18-FEB-92
  106-   111 |                     18  2 92 | Date of last update: 18-FEB-92
             |                             |
  112-   113 |                         709 | Number of characters
  114-   115 |                        1000 | Bodysize
  116-   117 |                         667 | Capital height
  118-   119 |                         467 | X-height
  120-   121 |                        -217 | Distance base line-lower body line
  122-   123 |                           0 | Italic angle
  124-   125 |                        1000 | Em-resolution in X-direction (in pixels)
  126-   127 |                        1000 | Em-resolution in Y-direction (in pixels)
  128-   129 |                           0 | X-minimum of typeface bounding box
  130-   131 |                        1163 | X-maximum of typeface bounding box
  132-   133 |                        -262 | Y-minimum of typeface bounding box
  134-   135 |                         892 | Y-maximum of typeface bounding box
             |                             |
  136-   137 |                           8 | Maximum number of serifs per character
  138-   139 |                          16 | Maximum number of peaks/character
  140-   141 |                          13 | Maximum number of curves/character
  142-   143 |                          12 | Max. number of horiz. curve extremes/character
  144-   145 |                          14 | Max. number of vert. curve extremes/character
  146-   147 |                          25 | Max. number of contours/character
             |                             |
  148-   149 |                           0 | Number of additional invisible horizontal lines
             |                             |
  150-   151 |                           0 | Total number of invisible vertical lines
--------------|----------------------------|------------------------------------------------
      CHARACTER DIRECTORY                  |
      -------------------                  |
    0-     1 |                        5676 | Length of character directory (in bytes)
             |                             |
    2-     3 |                         101 | Number of 1. character
    4-     7 |                        5828 | Pointer to 1. character
    8-     9 |                         151 | Length of 1. character (in bytes)
             |                             |
   10-    17 |     102       5980      169 | No., pointer, length of     2. character
   18-    25 |     103       6150      132 | No., pointer, length of     3. character
   26-    33 |     104       6282      122 | No., pointer, length of     4. character
   34-    41 |     105       6404      178 | No., pointer, length of     5. character
   42-    49 |     106       6582      162 | No., pointer, length of     6. character
   50-    57 |     107       6744      186 | No., pointer, length of     7. character
   58-    65 |     108       6930      208 | No., pointer, length of     8. character
   66-    73 |     109       7138      112 | No., pointer, length of     9. character
   74-    81 |     110       7250       96 | No., pointer, length of    10. character
   82-    89 |     111       7346      208 | No., pointer, length of    11. character
   90-    97 |     112       7554      119 | No., pointer, length of    12. character
   98-   105 |     113       7674      171 | No., pointer, length of    13. character
  106-   113 |     114       7846      138 | No., pointer, length of    14. character
  114-   121 |     115       7984       97 | No., pointer, length of    15. character
--------------|----------------------------|------------------------------------------------
```



```
U003003T.BS                         18-FEB-92                              PAGE 14

--------------|----------------------------|------------------------------------------------
   BYTENO.    |          CONTENT           |                  COMMENT
--------------|----------------------------|------------------------------------------------
    CHARACTER DATA                         |
    --------------                         |
    CHARACTER HEADER                       |
    ----------------                       |
   0-       1 |                          1 | Control word: Upper case character
   2-       3 |                         63 | Number of digitizations
   4-       5 |                        549 | Total width
   6-       7 |                         53 | Left side bearing
   8-       9 |                        487 | Width
  10-      11 |                        -10 | Y-minimum of digitizations
  12-      13 |                        477 | Y-maximum of digitizations
--------------|----------------------------|------------------------------------------------
    STEM BLOCK                             |
    ----------                             |
  14-      15 |                         10 | Length of stem block (in bytes)
              |                            |
  16          |                          2 | Number of contours for  1. contour group
              |                            |
  17          |                          2 | Number of visible vertical straight lines
  18-      19 |       44                   | Index            (absolute stem, black)
  20-      22 |       61                44 | Index, reference (relative stem, white)
              |                            |
  23          |                          0 | Number of horizontal straight lines
--------------|----------------------------|------------------------------------------------
    CONTOUR INDEX                          |
    -------------                          |
  24-      25 |                          2 | Length of contour index (in bytes)
--------------|----------------------------|------------------------------------------------
    IMAGE INFORMATION                      |
    -----------------                      |
  26-      27 |                        184 | Length of image information (in bytes)
              |                            |
  28-      32 | 01000100      475       69 |   475       69  (X,Y)       1. DIG: W-Vector
  33          | 00010110                   | Vert. peak               black
  34-      36 | 01100000       12      -16 |   487       53  (X,Y)       2. DIG
  37          | 00010010                   | Horiz. peak              black
  38-      40 | 10110000      -44      -43 |   443       10  (X,Y)       3. DIG
  41-      43 | 10110000      -31      -17 |   412       -7  (X,Y)       4. DIG
  44-      45 | 01010010      -35        0 |   377       -7  (X,Y)       5. DIG
  46-      47 | 01001001               255 | Rel. Horiz. Curv. Extr. white   Baseline
  48-      49 | 10010010      -41        0 |   336       -7  (X,Y)       6. DIG
  50-      52 | 10010000      -23       24 |   313       17  (X,Y)       7. DIG
  53-      55 | 01010000      -14       59 |   299       76  (X,Y)       8. DIG
  56          | 00010101                   | Vert. peak               white
  57-      59 | 10110000      -54      -61 |   245       15  (X,Y)       9. DIG
  60-      62 | 10110000      -47      -25 |   198      -10  (X,Y)      10. DIG
  63-      64 | 01010010      -62        0 |   136      -10  (X,Y)      11. DIG
  65-      66 | 01001001               255 | Rel. Horiz. Curv. Extr. white   Baseline
  67-      68 | 10010010      -82        0 |    54      -10  (X,Y)      12. DIG
  69-      71 | 10010000      -54       53 |     0       43  (X,Y)      13. DIG
  72-      73 | 01000001        0       82 |     0      125  (X,Y)      14. DIG
  74          | 01000101                   | Abs. Vert. Curv. Extr.  white
  75-      76 | 10000001        0       84 |     0      209  (X,Y)      15. DIG
  77-      79 | 10000000       44       37 |    44      246  (X,Y)      16. DIG
  80-      82 | 00000000      124       21 |   168      267  (X,Y)      17. DIG
  83-      85 | 10000000       16        2 |   184      269  (X,Y)      18. DIG
--------------|----------------------------|------------------------------------------------
```

```
U003003T.BS                            18-FEB-92                              PAGE 16

-------------|----------------------------|-------------------------------------------
   BYTENO.   |          CONTENT           |                   COMMENT
-------------|----------------------------|-------------------------------------------
     CHARACTER HEADER                     |
     ----------------                     |
   0-      1 |                          1 | Control word: Upper case character
   2-      3 |                         41 | Number of digitizations
   4-      5 |                        628 | Total width
   6-      7 |                         -3 | Left side bearing
   8-      9 |                        586 | Width
  10-     11 |                        -13 | Y-minimum of digitizations
  12-     13 |                        690 | Y-maximum of digitizations
-------------|----------------------------|-------------------------------------------
     STEM BLOCK                           |
     ----------                           |
  14-     15 |                         12 | Length of stem block (in bytes)
             |                            |
  16         |                          2 | Number of contours for  1. contour group
             |                            |
  17         |                          2 | Number of visible vertical straight lines
  18-     19 |         1                  | Index             (absolute stem, white)
  20-     22 |         8                1 | Index, reference (relative stem, black)
             |                            |
  23         |                          1 | Number of horizontal straight lines
  24-     25 |        27                  | Index             (absolute stem, white)
-------------|----------------------------|-------------------------------------------
     CONTOUR INDEX                        |
     -------------                        |
  26-     27 |                          2 | Length of contour index (in bytes)
-------------|----------------------------|-------------------------------------------
     IMAGE INFORMATION                    |
     -----------------                    |
  28-     29 |                        118 | Length of image information (in bytes)
             |                            |
  30-     33 | 00100000       127     -13 |   399       147  (X,Y)       1. DIG: W-Vector
  34-     35 | 00010010       -25       0 |   374       147  (X,Y)       2. DIG
  36-     38 | 01001001         0     546 |   374       693  (X,Y)       3. DIG
  39         | 00100100                   | End of serif            grey
  40-     41 | 10000001         0      87 |   374       780  (X,Y)       4. DIG
  42-     44 | 10010000       -13      15 |   361       795  (X,Y)       5. DIG
  45-     47 | 00010000       -74       3 |   287       798  (X,Y)       6. DIG
  48-     49 | 00010010       -15       0 |   272       798  (X,Y)       7. DIG
  50-     51 | 01000001         0      29 |   272       827  (X,Y)       8. DIG
  52-     53 | 00101010                254 | Start of serif          black   Cap height
  54-     56 | 00000000       189      23 |   461       850  (X,Y)       9. DIG
  57-     59 | 00101001         0    -296 |   461       554  (X,Y)      10. DIG
  60-     62 | 10000000        47      60 |   508       614  (X,Y)      11. DIG
  63-     65 | 10000000        46      23 |   554       637  (X,Y)      12. DIG
  66-     67 | 01000010        73       0 |   627       637  (X,Y)      13. DIG
  68-     69 | 01001010                253 | Rel. Horiz. Curv. Extr. black   X-height
  70-     71 | 10000010       141       0 |   768       637  (X,Y)      14. DIG
  72-     74 | 10100000        90     -94 |   858       543  (X,Y)      15. DIG
  75-     76 | 01100001         0    -147 |   858       396  (X,Y)      16. DIG
  77         | 01000110                   | Abs. Vert. Curv. Extr.  black
  78-     79 | 10100001         0    -154 |   858       242  (X,Y)      17. DIG
  80-     82 | 10110000       -91     -92 |   767       150  (X,Y)      18. DIG
  83-     84 | 01010010      -151       0 |   616       150  (X,Y)      19. DIG
  85-     86 | 01001001                255 | Rel. Horiz. Curv. Extr. white   Baseline
  87-     88 | 10010010       -66       0 |   550       150  (X,Y)      20. DIG
-------------|----------------------------|-------------------------------------------
```

```
U003003T.BS                             18-FEB-92                                    PAGE 18

--------------|-----------------------------|-------------------------------------------------
    BYTENO.   |          CONTENT            |                    COMMENT
--------------|-----------------------------|-------------------------------------------------
      CHARACTER HEADER                      |
      ----------------                      |
    0-      1 |                           1 | Control word: Upper case character
    2-      3 |                          38 | Number of digitizations
    4-      5 |                         495 | Total width
    6-      7 |                          45 | Left side bearing
    8-      9 |                         430 | Width
   10-     11 |                         -10 | Y-minimum of digitizations
   12-     13 |                         477 | Y-maximum of digitizations
--------------|-----------------------------|-------------------------------------------------
      STEM BLOCK                            |
      ----------                            |
   14-     15 |                           4 | Length of stem block (in bytes)
              |                             |
   16         |                          -1 | Number of contours for  1. contour group
              |                             |
              |                           0 | Number of visible vertical straight lines
              |                           0 | Number of visible horizontal straight lines
--------------|-----------------------------|-------------------------------------------------
      CONTOUR INDEX                         |
      -------------                         |
   18-     19 |                           2 | Length of contour index (in bytes)
--------------|-----------------------------|-------------------------------------------------
      IMAGE INFORMATION                     |
      -----------------                     |
   20-     21 |                         113 | Length of image information (in bytes)
              |                             |
   22-     26 | 01000100       405      103 |   1153       756  (X,Y)       1. DIG: W-Vector
   27         | 00010110                    | Vert. peak              black
   28-     30 | 01100000        22      -22 |   1175       734  (X,Y)       2. DIG
   31         | 00010010                    | Horiz. peak             black
   32-     34 | 10110000       -64      -68 |   1111       666  (X,Y)       3. DIG
   35-     37 | 10110000       -51      -23 |   1060       643  (X,Y)       4. DIG
   38-     39 | 01010010       -80        0 |    980       643  (X,Y)       5. DIG
   40-     41 | 01001001                255 | Rel. Horiz. Curv. Extr. white   Baseline
   42-     43 | 10010010      -142        0 |    838       643  (X,Y)       6. DIG
   44-     46 | 10010000       -90       95 |    748       738  (X,Y)       7. DIG
   47-     48 | 01000001         0      149 |    748       887  (X,Y)       8. DIG
   49         | 01000101                    | Abs. Vert. Curv. Extr.  white
   50-     51 | 10000001         0      154 |    748      1041  (X,Y)       9. DIG
   52-     54 | 10000000        90       89 |    838      1130  (X,Y)      10. DIG
   55-     56 | 01000010       154        0 |    992      1130  (X,Y)      11. DIG
   57-     58 | 01001010                253 | Rel. Horiz. Curv. Extr. black   X-height
   59-     60 | 10000010       108        0 |   1100      1130  (X,Y)      12. DIG
   61-     63 | 10100000        78      -48 |   1178      1082  (X,Y)      13. DIG
   64-     65 | 01100001         0      -67 |   1178      1015  (X,Y)      14. DIG
   66         | 01000110                    | Abs. Vert. Curv. Extr.  black
   67-     68 | 10100001         0      -34 |   1178       981  (X,Y)      15. DIG
   69-     71 | 10110000       -25      -24 |   1153       957  (X,Y)      16. DIG
   72-     73 | 01010010       -36        0 |   1117       957  (X,Y)      17. DIG
   74         | 01000001                    | Abs. Horiz. Curv. Extr. white
   75-     76 | 10010010       -32        0 |   1085       957  (X,Y)      18. DIG
   77-     79 | 10010000       -23       21 |   1062       978  (X,Y)      19. DIG
   80-     81 | 01000001         0       30 |   1062      1008  (X,Y)      20. DIG
   82         | 01000101                    | Abs. Vert. Curv. Extr.  white
   83-     84 | 10000001         0       18 |   1062      1026  (X,Y)      21. DIG
--------------|-----------------------------|-------------------------------------------------
```

```
U003003T.BS                           18-FEB-92                          PAGE 20
--------------------------------------------------------------------------------


                                   GRAND TOTAL
                                   -----------
                                                               KBYTES
                                                               ------
                Length of font:                               113.662
                Length of fontheader:                           0.152
                Length of all character headers:                9.926
                Length of total image information:             84.433

                DIGITIZATIONS                                  NUMBER
                -----------------                              ------
                Serifs (start):                                  1152
                Serifs (end):                                     726
                Horizontal peaks:                                 838
                Vertikal peaks:                                   699
                Horizontal curve extremes:                       2402
                Vertical Curve extremes:                         2149
                Other digitizations:                            21130
                                                               ======
                Total:                                          29096
                -----------------------------------------------------------
                Characters:                                       709
                Size (in kbytes):                             107.507
                -----------------------------------------------------------
                Raster (grid):                            1000 x 1000
                -----------------------------------------------------------
                Digitizations/character:                           41
                Image information/character:                      119 bytes


--------------------------------------------------------------------------------
```

Description of the VE format

1. Structure of a VE font

For the structure of a VE font see Appendices G and L, extended structure.

Like the VC format, the VE format has a (possibly) limited resolution (lower than 15,000 / em).

2. Description of the VE format

The **vector format** is a simplified form of the VC format. Arcs are no longer described as circles but are divided into vectors and coded.

The curved sections of an IK type font comprise almost 90% of the image information. As already mentioned (Chapter 5.1.4), raster images seldom have long, straight sections within curved parts comprising more than 13 contour pixels.

This fact has led to an incremental coding in which an 8 bit byte is sufficient for a vector. Such vectors are known as V vectors. The left 4 bits (higher value) describes the X increment of the V vector, the right 4 bits (lower value) the Y increment.

The amounts Δ X and Δ Y of the V vectors are kept smaller than 14. The sign is determined specially for each case.

As in the case of a VC format, there are also white vectors running to the start of contours with pen «raised». They begin on the word boundary at the beginning of the contour data. The associated Δ X and Δ Y values are signed 16 bit integers. The first white vector starts at the zero point of the appropriate coordinate system of the character. Subsequent white vectors start at the ends of the preceding contours and lead to the following contour in a sequence given by the contour index.

Longer straights cannot, of course, be described using V vectors, and for this reason we have introduced the following system (hexadecimal notation).

The vectors E0...ED indicate that the Y component is taken from the least significant 4 bits while the X component is described in the following byte. In doing so, the byte values 1...120 are taken as positive and the values 255...136 counted as negative, as -1...-120. We call this the S vector method.

From this point X components of vectors have the sign given here. Vectors F0...FD are, in effect, the same and additionally change the sign of the following Y components.

If the following byte is coded with 80, the coding, as signed integer, follows in the next 2 bytes (16 bit). We call this the L vector method. The sign of the components also applies to the subsequent V vectors.

Correspondingly the vectors 0E...DE and 0F...DF mean that Y components are described in the following byte or following 3 bytes. Byte EE, EF, FE or FF mean that first the X, and then Y components are described as discussed previously.

The switch-over of the sign for short increments Δ X and Δ Y of the V vectors is carried out by calling up the longer code (either 8 or 16 bit) for one or, in some cases, both components. The stating of F0...FD or 0F...DF allows simultaneous switching of the sign of the other respective component. This switchover first applies, however, from the next V vector onwards and not for the vector currently being dealt with.

Coding examples: sequence of increments

Increment	Hexadecimal (INTEGER × 2)	Status of sign XY	Comments
5,5	55	+ +	Normal V vector
– 10,0	E0,F6	– +	Sign change in X
– 14,10	EA,F2	– +	S vector method for X
– 10,14	AE,0E	– +	S vector method for Y
– 1000,0	E0,80,FC18	– +	L vector method for X
– 10, – 1000	AF,80,FC18	– –	L vector method for Y
		+ –	Sign change for X
10,10	AE,0A	+ +	Sign change for Y

Increment	Hexadecimal (INTEGER × 2)	Status of sign XY	Comments
200,200	FF,80,00C8	+ +	L method for X and Y with double sign change
	80,00C8	– –	
–10,–10	AA		
5,–5	F5,05	+ –	Sign change in X
		+ +	Sign change in Y
4,4	4F,04		
–3,3	33	– +	Sign change in X

The switch-over of the other component respectively, in the case of longer components, is particularly very favourable when digitizing curves because at the extremes, one component is long whereas the other is short and often changes its sign.

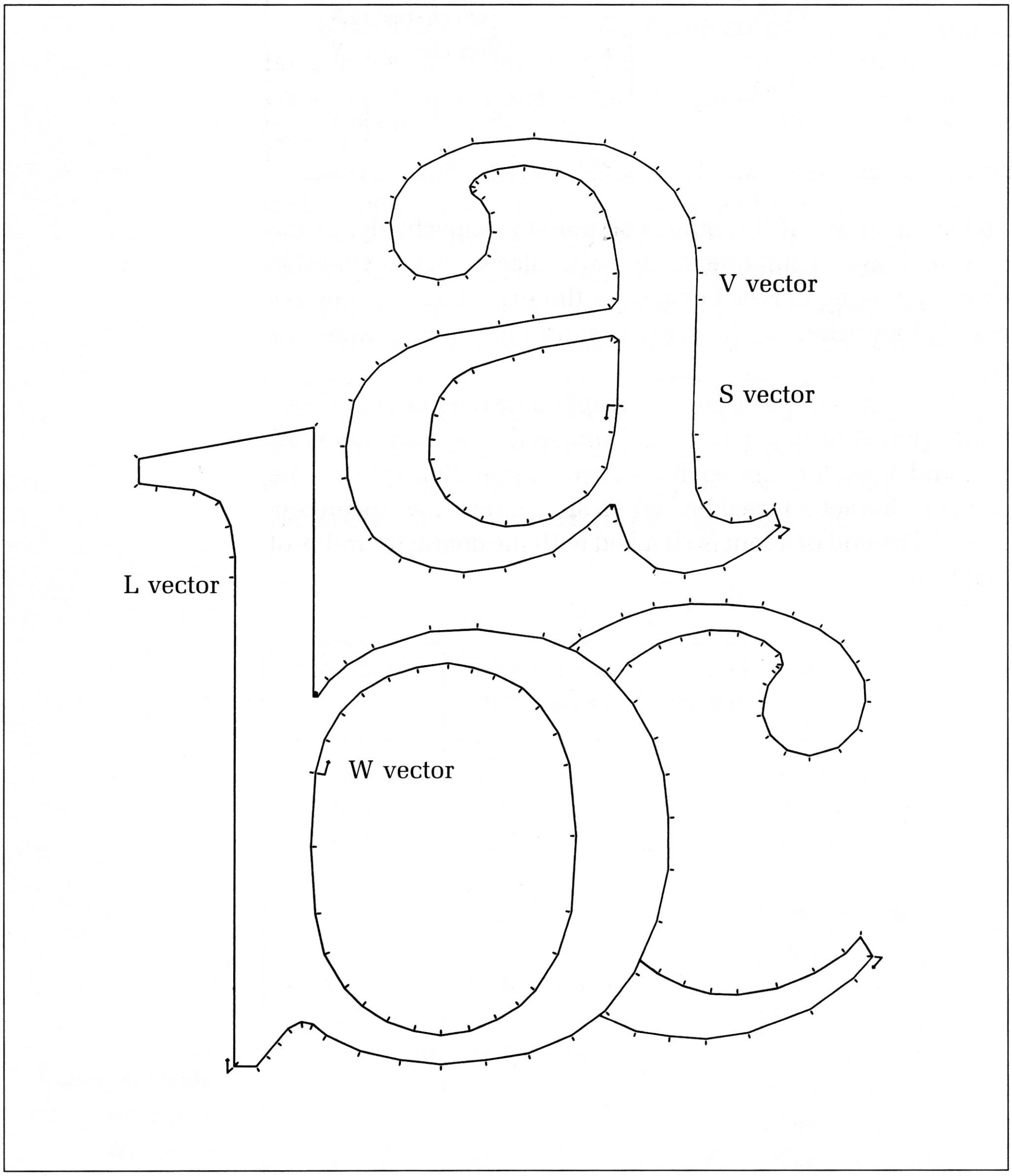
The end of the contour is controlled by the number of vectors (digitizations), details to be found in the contour index. W, S, L and V vectors are each counted as one digitization. The end of a character is reached when all contours have been processed.The end of a font is checked with the character index of that font.

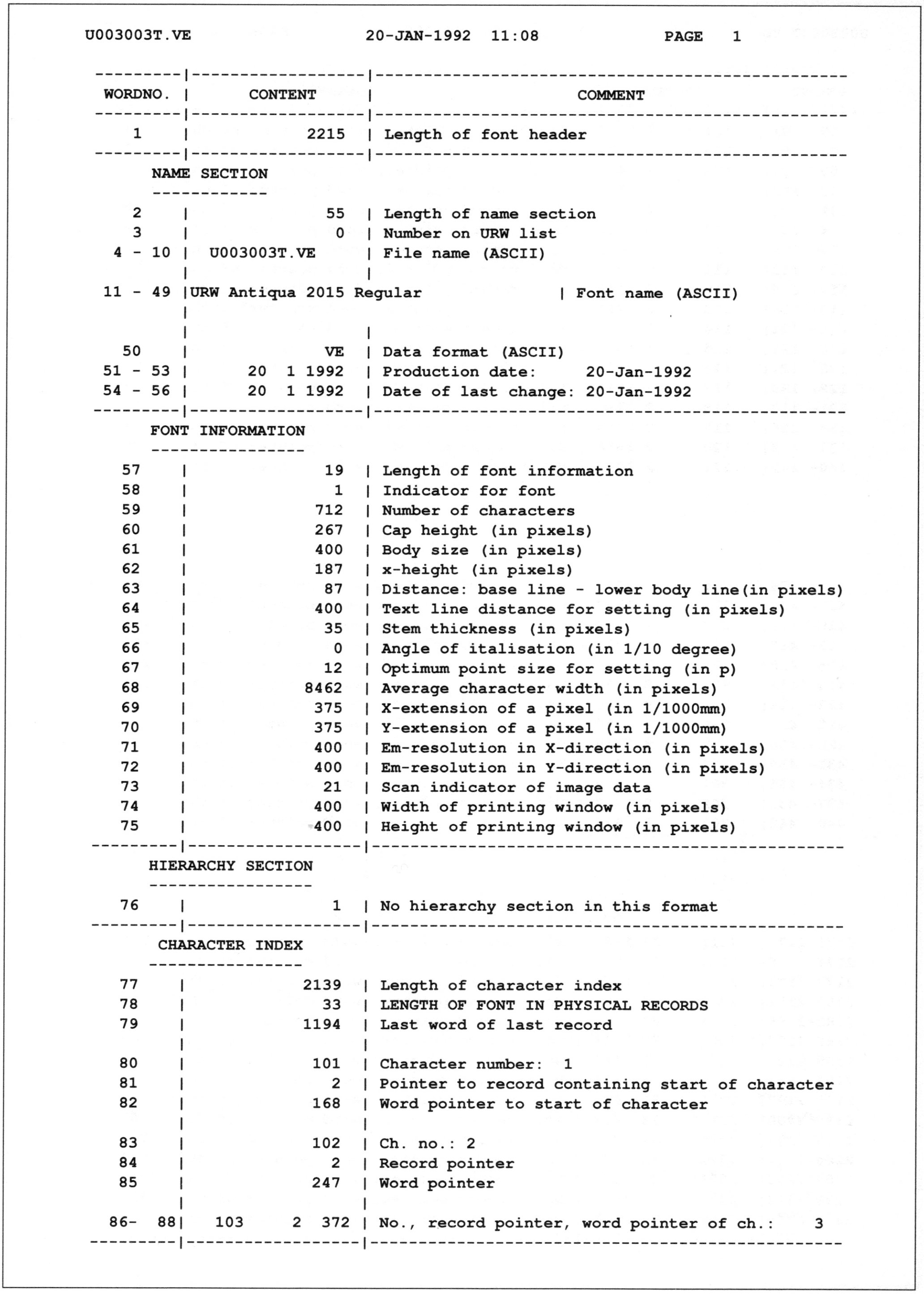
3. Example of a VE format

First of all we represent the lower case letters a, b, c of the typeface «URW Antiqua 2015 normal», with marks.

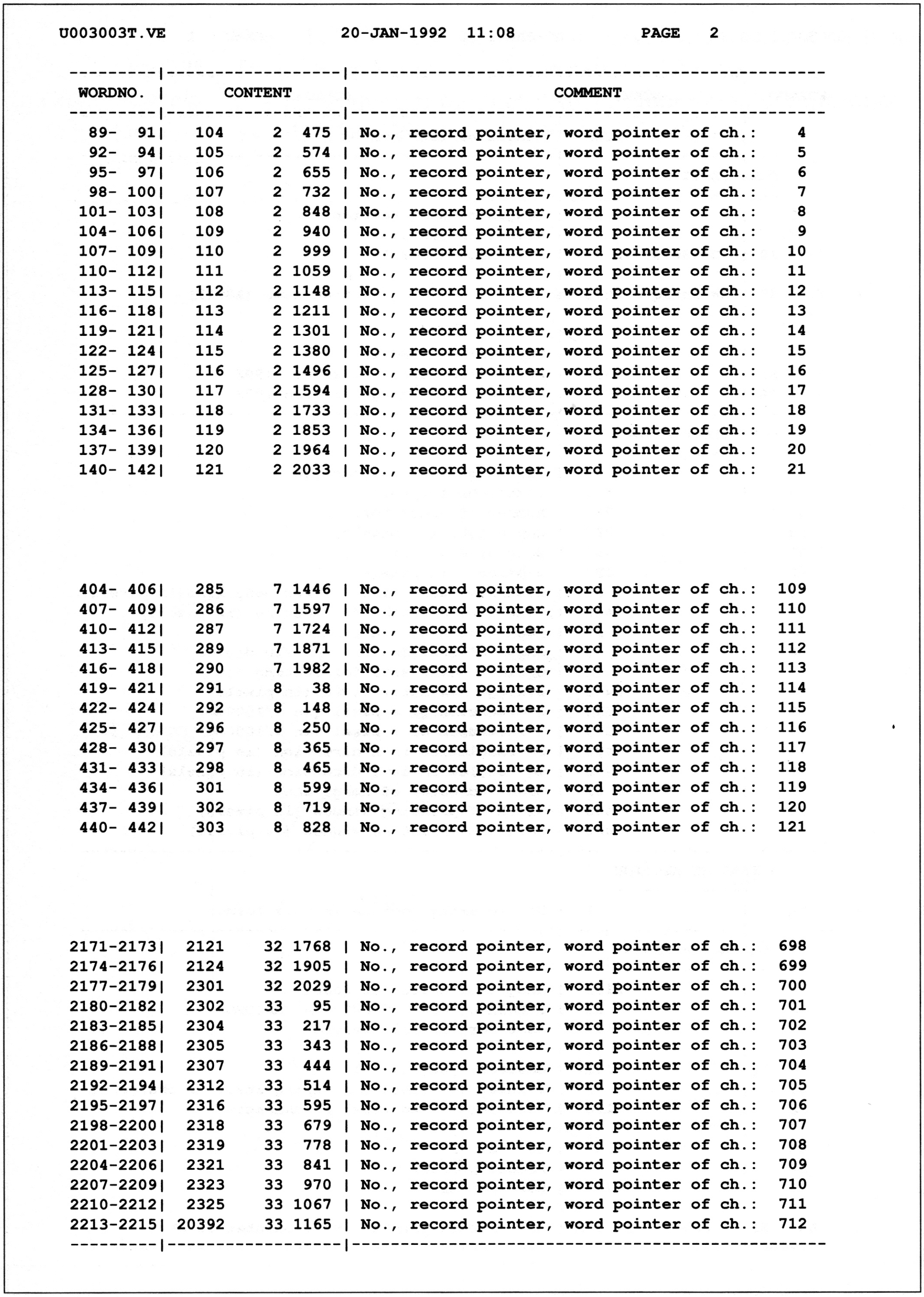
Subsequently the complete type font format is reproduced as a data list. The raster is 400 × 400 for the em.

Fig. 196
Coding example: sequence of increments





```
U003003T.VE                 20-JAN-1992  11:08            PAGE   1

 ---------|------------------|------------------------------------------------
  WORDNO. |     CONTENT      |                     COMMENT
 ---------|------------------|------------------------------------------------
     1    |              2215 | Length of font header
 ---------|------------------|------------------------------------------------
       NAME SECTION
       ------------
     2    |                55 | Length of name section
     3    |                 0 | Number on URW list
   4 - 10 |  U003003T.VE      | File name (ASCII)
          |                   |
  11 - 49 |URW Antiqua 2015 Regular               | Font name (ASCII)
          |
          |                   |
    50    |                VE | Data format (ASCII)
  51 - 53 |      20  1 1992   | Production date:     20-Jan-1992
  54 - 56 |      20  1 1992   | Date of last change: 20-Jan-1992
 ---------|------------------|------------------------------------------------
       FONT INFORMATION
       ----------------
    57    |                19 | Length of font information
    58    |                 1 | Indicator for font
    59    |               712 | Number of characters
    60    |               267 | Cap height (in pixels)
    61    |               400 | Body size (in pixels)
    62    |               187 | x-height (in pixels)
    63    |                87 | Distance: base line - lower body line(in pixels)
    64    |               400 | Text line distance for setting (in pixels)
    65    |                35 | Stem thickness (in pixels)
    66    |                 0 | Angle of italisation (in 1/10 degree)
    67    |                12 | Optimum point size for setting (in p)
    68    |              8462 | Average character width (in pixels)
    69    |               375 | X-extension of a pixel (in 1/1000mm)
    70    |               375 | Y-extension of a pixel (in 1/1000mm)
    71    |               400 | Em-resolution in X-direction (in pixels)
    72    |               400 | Em-resolution in Y-direction (in pixels)
    73    |                21 | Scan indicator of image data
    74    |               400 | Width of printing window (in pixels)
    75    |               400 | Height of printing window (in pixels)
 ---------|------------------|------------------------------------------------
       HIERARCHY SECTION
       -----------------
    76    |                 1 | No hierarchy section in this format
 ---------|------------------|------------------------------------------------
        CHARACTER INDEX
       ----------------
    77    |              2139 | Length of character index
    78    |                33 | LENGTH OF FONT IN PHYSICAL RECORDS
    79    |              1194 | Last word of last record
          |                   |
    80    |               101 | Character number:  1
    81    |                 2 | Pointer to record containing start of character
    82    |               168 | Word pointer to start of character
          |                   |
    83    |               102 | Ch. no.: 2
    84    |                 2 | Record pointer
    85    |               247 | Word pointer
          |                   |
  86-  88|   103     2  372 | No., record pointer, word pointer of ch.:    3
 ---------|------------------|------------------------------------------------
```



```
U003003T.VE                 20-JAN-1992  11:08             PAGE    2

---------|------------------|------------------------------------------
 WORDNO. |     CONTENT      |                 COMMENT
---------|------------------|------------------------------------------
  89-  91|  104      2  475 | No., record pointer, word pointer of ch.:    4
  92-  94|  105      2  574 | No., record pointer, word pointer of ch.:    5
  95-  97|  106      2  655 | No., record pointer, word pointer of ch.:    6
  98- 100|  107      2  732 | No., record pointer, word pointer of ch.:    7
 101- 103|  108      2  848 | No., record pointer, word pointer of ch.:    8
 104- 106|  109      2  940 | No., record pointer, word pointer of ch.:    9
 107- 109|  110      2  999 | No., record pointer, word pointer of ch.:   10
 110- 112|  111      2 1059 | No., record pointer, word pointer of ch.:   11
 113- 115|  112      2 1148 | No., record pointer, word pointer of ch.:   12
 116- 118|  113      2 1211 | No., record pointer, word pointer of ch.:   13
 119- 121|  114      2 1301 | No., record pointer, word pointer of ch.:   14
 122- 124|  115      2 1380 | No., record pointer, word pointer of ch.:   15
 125- 127|  116      2 1496 | No., record pointer, word pointer of ch.:   16
 128- 130|  117      2 1594 | No., record pointer, word pointer of ch.:   17
 131- 133|  118      2 1733 | No., record pointer, word pointer of ch.:   18
 134- 136|  119      2 1853 | No., record pointer, word pointer of ch.:   19
 137- 139|  120      2 1964 | No., record pointer, word pointer of ch.:   20
 140- 142|  121      2 2033 | No., record pointer, word pointer of ch.:   21





 404- 406|  285      7 1446 | No., record pointer, word pointer of ch.:  109
 407- 409|  286      7 1597 | No., record pointer, word pointer of ch.:  110
 410- 412|  287      7 1724 | No., record pointer, word pointer of ch.:  111
 413- 415|  289      7 1871 | No., record pointer, word pointer of ch.:  112
 416- 418|  290      7 1982 | No., record pointer, word pointer of ch.:  113
 419- 421|  291      8   38 | No., record pointer, word pointer of ch.:  114
 422- 424|  292      8  148 | No., record pointer, word pointer of ch.:  115
 425- 427|  296      8  250 | No., record pointer, word pointer of ch.:  116
 428- 430|  297      8  365 | No., record pointer, word pointer of ch.:  117
 431- 433|  298      8  465 | No., record pointer, word pointer of ch.:  118
 434- 436|  301      8  599 | No., record pointer, word pointer of ch.:  119
 437- 439|  302      8  719 | No., record pointer, word pointer of ch.:  120
 440- 442|  303      8  828 | No., record pointer, word pointer of ch.:  121





2171-2173| 2121     32 1768 | No., record pointer, word pointer of ch.:  698
2174-2176| 2124     32 1905 | No., record pointer, word pointer of ch.:  699
2177-2179| 2301     32 2029 | No., record pointer, word pointer of ch.:  700
2180-2182| 2302     33   95 | No., record pointer, word pointer of ch.:  701
2183-2185| 2304     33  217 | No., record pointer, word pointer of ch.:  702
2186-2188| 2305     33  343 | No., record pointer, word pointer of ch.:  703
2189-2191| 2307     33  444 | No., record pointer, word pointer of ch.:  704
2192-2194| 2312     33  514 | No., record pointer, word pointer of ch.:  705
2195-2197| 2316     33  595 | No., record pointer, word pointer of ch.:  706
2198-2200| 2318     33  679 | No., record pointer, word pointer of ch.:  707
2201-2203| 2319     33  778 | No., record pointer, word pointer of ch.:  708
2204-2206| 2321     33  841 | No., record pointer, word pointer of ch.:  709
2207-2209| 2323     33  970 | No., record pointer, word pointer of ch.:  710
2210-2212| 2325     33 1067 | No., record pointer, word pointer of ch.:  711
2213-2215| 20392    33 1165 | No., record pointer, word pointer of ch.:  712
---------|------------------|------------------------------------------
```

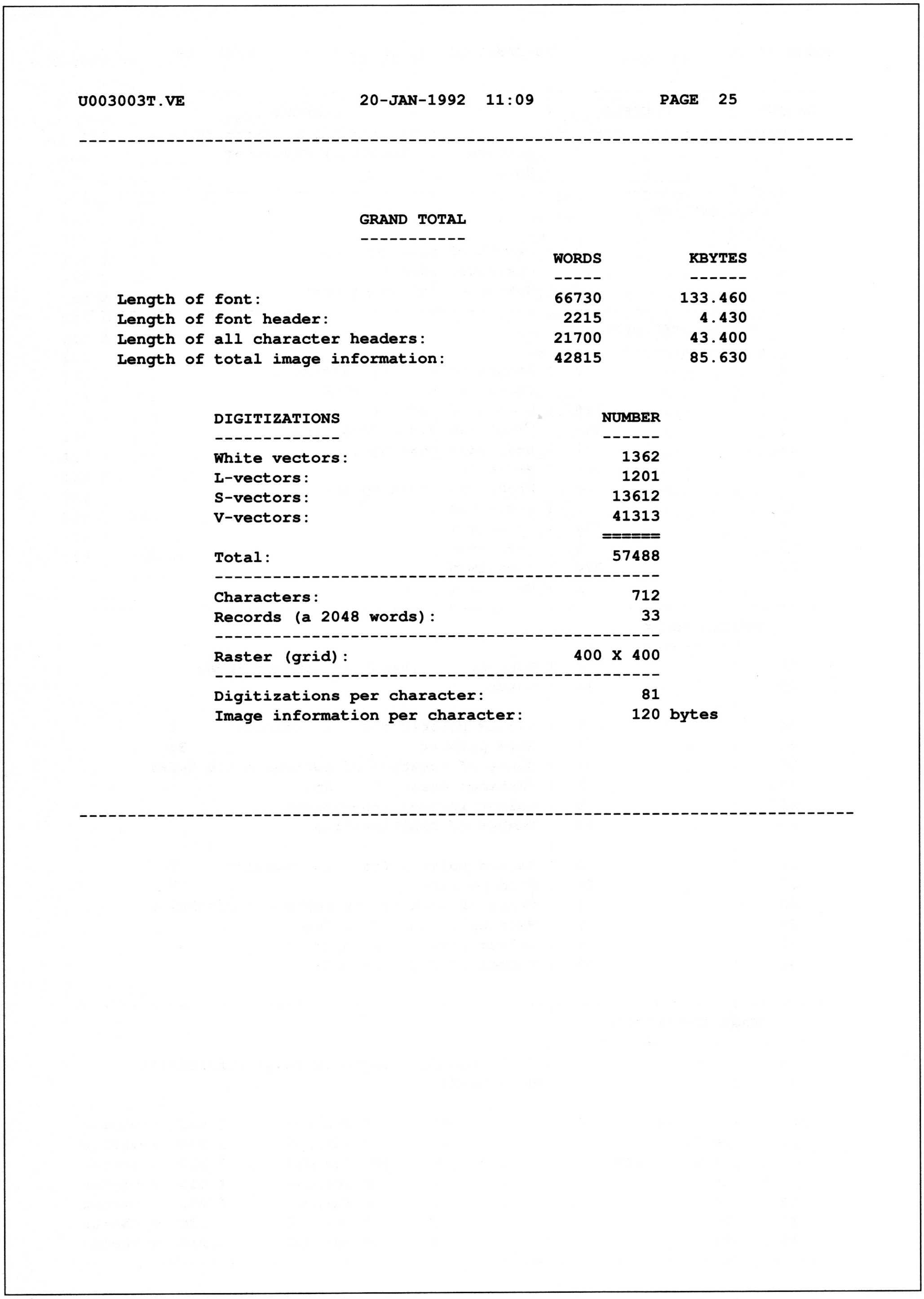
```
U003003T.VE                 20-JAN-1992  11:08             PAGE  15

---------|------------------|-------------------------------------------------
 WORDNO. |      CONTENT     |                    COMMENT
---------|------------------|-------------------------------------------------
    1    |                0 | Records      length of character
    2    |              120 | Words
---------|------------------|-------------------------------------------------
      NAME SECTION
      ------------
    3    |                3 | Length of name section
    4    |              301 | Character number
    5    |                0 | Number of following part
---------|------------------|-------------------------------------------------
      SETTING INFORMATION
      -------------------
    6    |               12 | Length of setting information
    7    |                1 | Character type = letter
    8    |              140 | Number of vectors
    9    |              220 | Total set width (T=L+W+R)
   10    |               21 | Left side bearing (L)
   11    |              194 | Width (W)
   12    |                5 | Right side bearing (R)
   13    |                0 | X-minimum
   14    |              194 | X-maximum
   15    |               -4 | Y-minimum
   16    |              191 | Y-maximum
   17    |                1 | Unit is 1 pixel
---------|------------------|-------------------------------------------------
      CONTOUR INDEX
      -------------
   18    |                0 | Records     length of contour index
   19    |               14 | Words
         |                  |
   20    |                0 | Record pointer for   1. contour:     0
   21    |                3 | Word pointer                   :    34
   22    |               -1 | Sense of rotation of contour = clockwise
   23    |                0 | Nesting: outer  1. order
   24    |                0 | Colour inside: transparent
   25    |              108 | Number of digitizations
         |                  |
   26    |                0 | Record pointer for   2. contour:     0
   27    |               70 | Word pointer                   :   101
   28    |               -1 | Sense of rotation of contour = clockwise
   29    |                1 | Nesting: inner  1. order
   30    |                0 | Colour inside: transparent
   31    |               32 | Number of digitizations
         |                  |
---------|------------------|-------------------------------------------------
      IMAGE INFORMATION
      -----------------
   32    |                     0  | Records  length of image information
   33    |                    89  | Words
         |                       |
  34- 100|     190      28       |    190      28 (dX,dY)      1.DIG  w-vector
   36 1  |4F F9                  |      4      -7 (dx,dY)      2.DIG  s-vector
   37 1  |99                     |     -9      -9 (dx,dy)      3.DIG  v-vector
   37 2  |86                     |     -8      -6 (dx,dy)      4.DIG  v-vector
   38 1  |84                     |     -8      -4 (dx,dy)      5.DIG  v-vector
   38 2  |A4                     |    -10      -4 (dx,dy)      6.DIG  v-vector
   39 1  |F1 F8                  |     -8      -1 (dX,dy)      7.DIG  s-vector
---------|-----------------------|---------------------------------------------
```

```
U003003T.VE                20-JAN-1992  11:08            PAGE  16

 ---------|------------------------|------------------------------------------
  WORDNO. |        CONTENT         |                COMMENT
 ---------|------------------------|------------------------------------------
    40 1  |61                      |     -6       1 (dx,dy)       8.DIG  v-vector
    40 2  |62                      |     -6       2 (dx,dy)       9.DIG  v-vector
    41 1  |74                      |     -7       4 (dx,dy)      10.DIG  v-vector
    41 2  |45                      |     -4       5 (dx,dy)      11.DIG  v-vector
    42 1  |46                      |     -4       6 (dx,dy)      12.DIG  v-vector
    42 2  |26                      |     -2       6 (dx,dy)      13.DIG  v-vector
    43 1  |F9 FE                   |     -2       9 (dX,dy)      14.DIG  s-vector
    44 1  |CC                      |    -12     -12 (dx,dy)      15.DIG  v-vector
    44 2  |76                      |     -7      -6 (dx,dy)      16.DIG  v-vector
    45 1  |85                      |     -8      -5 (dx,dy)      17.DIG  v-vector
    45 2  |A5                      |    -10      -5 (dx,dy)      18.DIG  v-vector
    46 1  |A4                      |    -10      -4 (dx,dy)      19.DIG  v-vector
    46 2  |A1                      |    -10      -1 (dx,dy)      20.DIG  v-vector
    47 1  |F1 F7                   |     -9      -1 (dX,dy)      21.DIG  s-vector
    48 1  |71                      |     -7       1 (dx,dy)      22.DIG  v-vector
    48 2  |81                      |     -8       1 (dx,dy)      23.DIG  v-vector
    49 1  |72                      |     -7       2 (dx,dy)      24.DIG  v-vector
    49 2  |62                      |     -6       2 (dx,dy)      25.DIG  v-vector
    50 1  |86                      |     -8       6 (dx,dy)      26.DIG  v-vector
    50 2  |44                      |     -4       4 (dx,dy)      27.DIG  v-vector
    51 1  |56                      |     -5       6 (dx,dy)      28.DIG  v-vector
    51 2  |46                      |     -4       6 (dx,dy)      29.DIG  v-vector
    52 1  |26                      |     -2       6 (dx,dy)      30.DIG  v-vector
    52 2  |27                      |     -2       7 (dx,dy)      31.DIG  v-vector
    53 1  |1F 0D                   |     -1      13 (dx,dY)      32.DIG  s-vector
    54 1  |1B                      |      1      11 (dx,dy)      33.DIG  v-vector
    54 2  |3A                      |      3      10 (dx,dy)      34.DIG  v-vector
    55 1  |25                      |      2       5 (dx,dy)      35.DIG  v-vector
    55 2  |24                      |      2       4 (dx,dy)      36.DIG  v-vector
    56 1  |34                      |      3       4 (dx,dy)      37.DIG  v-vector
    56 2  |33                      |      3       3 (dx,dy)      38.DIG  v-vector
    57 1  |75                      |      7       5 (dx,dy)      39.DIG  v-vector
    57 2  |64                      |      6       4 (dx,dy)      40.DIG  v-vector
    58 1  |73                      |      7       3 (dx,dy)      41.DIG  v-vector
    58 2  |72                      |      7       2 (dx,dy)      42.DIG  v-vector
    59 1  |B3                      |     11       3 (dx,dy)      43.DIG  v-vector
    59 2  |E3 0F                   |     15       3 (dX,dy)      44.DIG  s-vector
    60 2  |E2 10                   |     16       2 (dX,dy)      45.DIG  s-vector
    61 2  |EF 25      07           |     37       7 (dX,dY)      46.DIG  s-vector
    63 1  |1E 10                   |     -1      16 (dx,dY)      47.DIG  s-vector
    64 1  |09                      |      0       9 (dx,dy)      48.DIG  v-vector
    64 2  |17                      |     -1       7 (dx,dy)      49.DIG  v-vector
    65 1  |4C                      |     -4      12 (dx,dy)      50.DIG  v-vector
    65 2  |35                      |     -3       5 (dx,dy)      51.DIG  v-vector
    66 1  |23                      |     -2       3 (dx,dy)      52.DIG  v-vector
    66 2  |43                      |     -4       3 (dx,dy)      53.DIG  v-vector
    67 1  |32                      |     -3       2 (dx,dy)      54.DIG  v-vector
    67 2  |A4                      |    -10       4 (dx,dy)      55.DIG  v-vector
    68 1  |F2 F0                   |    -16       2 (dX,dy)      56.DIG  s-vector
    69 1  |A1                      |    -10      -1 (dx,dy)      57.DIG  v-vector
    69 2  |82                      |     -8      -2 (dx,dy)      58.DIG  v-vector
    70 1  |63                      |     -6      -3 (dx,dy)      59.DIG  v-vector
    70 2  |43                      |     -4      -3 (dx,dy)      60.DIG  v-vector
    71 1  |22                      |     -2      -2 (dx,dy)      61.DIG  v-vector
    71 2  |0F FD                   |      0      -3 (dx,dY)      62.DIG  s-vector
    72 2  |55                      |      5      -5 (dx,dy)      63.DIG  v-vector
 ---------|------------------------|------------------------------------------
```

```
U003003T.VE                20-JAN-1992  11:08            PAGE  17

 ---------|-----------------------|-----------------------------------------
 WORDNO.  |        CONTENT        |                COMMENT
 ---------|-----------------------|-----------------------------------------
    73 1  |46                     |      4     -6 (dx,dy)      64.DIG  v-vector
    73 2  |13                     |      1     -3 (dx,dy)      65.DIG  v-vector
    74 1  |0F FC                  |      0     -4 (dx,dY)      66.DIG  s-vector
    75 1  |16                     |     -1     -6 (dx,dy)      67.DIG  v-vector
    75 2  |25                     |     -2     -5 (dx,dy)      68.DIG  v-vector
    76 1  |35                     |     -3     -5 (dx,dy)      69.DIG  v-vector
    76 2  |54                     |     -5     -4 (dx,dy)      70.DIG  v-vector
    77 1  |62                     |     -6     -2 (dx,dy)      71.DIG  v-vector
    77 2  |F0 F4                  |    -12      0 (dX,dy)      72.DIG  s-vector
    78 2  |42                     |     -4      2 (dx,dy)      73.DIG  v-vector
    79 1  |64                     |     -6      4 (dx,dy)      74.DIG  v-vector
    79 2  |46                     |     -4      6 (dx,dy)      75.DIG  v-vector
    80 1  |26                     |     -2      6 (dx,dy)      76.DIG  v-vector
    80 2  |0F 0B                  |      0     11 (dx,dY)      77.DIG  s-vector
    81 2  |26                     |      2      6 (dx,dy)      78.DIG  v-vector
    82 1  |35                     |      3      5 (dx,dy)      79.DIG  v-vector
    82 2  |34                     |      3      4 (dx,dy)      80.DIG  v-vector
    83 1  |77                     |      7      7 (dx,dy)      81.DIG  v-vector
    83 2  |64                     |      6      4 (dx,dy)      82.DIG  v-vector
    84 1  |53                     |      5      3 (dx,dy)      83.DIG  v-vector
    84 2  |E4 0E                  |     14      4 (dX,dy)      84.DIG  s-vector
    85 2  |E3 10                  |     16      3 (dX,dy)      85.DIG  s-vector
    86 2  |F1 11                  |     17      1 (dX,dy)      86.DIG  s-vector
    87 2  |E1 0E                  |     14     -1 (dX,dy)      87.DIG  s-vector
    88 2  |D2                     |     13     -2 (dx,dy)      88.DIG  v-vector
    89 1  |72                     |      7     -2 (dx,dy)      89.DIG  v-vector
    89 2  |62                     |      6     -2 (dx,dy)      90.DIG  v-vector
    90 1  |63                     |      6     -3 (dx,dy)      91.DIG  v-vector
    90 2  |43                     |      4     -3 (dx,dy)      92.DIG  v-vector
    91 1  |76                     |      7     -6 (dx,dy)      93.DIG  v-vector
    91 2  |34                     |      3     -4 (dx,dy)      94.DIG  v-vector
    92 1  |46                     |      4     -6 (dx,dy)      95.DIG  v-vector
    92 2  |35                     |      3     -5 (dx,dy)      96.DIG  v-vector
    93 1  |3B                     |      3    -11 (dx,dy)      97.DIG  v-vector
    93 2  |1D                     |      1    -13 (dx,dy)      98.DIG  v-vector
    94 1  |0E A7                  |      0    -89 (dx,dY)      99.DIG  s-vector
    95 1  |2C                     |      2    -12 (dx,dy)     100.DIG  v-vector
    95 2  |26                     |      2     -6 (dx,dy)     101.DIG  v-vector
    96 1  |34                     |      3     -4 (dx,dy)     102.DIG  v-vector
    96 2  |43                     |      4     -3 (dx,dy)     103.DIG  v-vector
    97 1  |F1 04                  |      4     -1 (dX,dy)     104.DIG  s-vector
    98 1  |51                     |      5      1 (dx,dy)     105.DIG  v-vector
    98 2  |52                     |      5      2 (dx,dy)     106.DIG  v-vector
    99 1  |64                     |      6      4 (dx,dy)     107.DIG  v-vector
    99 2  |43                     |      4      3 (dx,dy)     108.DIG  v-vector
  101- 120|     -70       76      |    -70     76 (dX,dY)     109.DIG  w-vector
   103 1  |0E D5                  |      0    -43 (dx,dY)     110.DIG  s-vector
   104 1  |2A                     |     -2    -10 (dx,dy)     111.DIG  v-vector
   104 2  |26                     |     -2     -6 (dx,dy)     112.DIG  v-vector
   105 1  |36                     |     -3     -6 (dx,dy)     113.DIG  v-vector
   105 2  |56                     |     -5     -6 (dx,dy)     114.DIG  v-vector
   106 1  |76                     |     -7     -6 (dx,dy)     115.DIG  v-vector
   106 2  |95                     |     -9     -5 (dx,dy)     116.DIG  v-vector
   107 1  |52                     |     -5     -2 (dx,dy)     117.DIG  v-vector
   107 2  |41                     |     -4     -1 (dx,dy)     118.DIG  v-vector
   108 1  |A1                     |    -10     -1 (dx,dy)     119.DIG  v-vector
 ---------|-----------------------|-----------------------------------------
```

```
U003003T.VE                 20-JAN-1992  11:08              PAGE  18

 ---------|-----------------------|------------------------------------------
  WORDNO. |        CONTENT        |                COMMENT
 ---------|-----------------------|------------------------------------------
   108 2  |F0 FB                  |     -5       0 (dX,dy)     120.DIG  s-vector
   109 2  |61                     |     -6       1 (dx,dy)     121.DIG  v-vector
   110 1  |52                     |     -5       2 (dx,dy)     122.DIG  v-vector
   110 2  |42                     |     -4       2 (dx,dy)     123.DIG  v-vector
   111 1  |77                     |     -7       7 (dx,dy)     124.DIG  v-vector
   111 2  |34                     |     -3       4 (dx,dy)     125.DIG  v-vector
   112 1  |23                     |     -2       3 (dx,dy)     126.DIG  v-vector
   112 2  |39                     |     -3       9 (dx,dy)     127.DIG  v-vector
   113 1  |18                     |     -1       8 (dx,dy)     128.DIG  v-vector
   113 2  |0F 05                  |      0       5 (dx,dY)     129.DIG  s-vector
   114 2  |17                     |      1       7 (dx,dy)     130.DIG  v-vector
   115 1  |14                     |      1       4 (dx,dy)     131.DIG  v-vector
   115 2  |24                     |      2       4 (dx,dy)     132.DIG  v-vector
   116 1  |35                     |      3       5 (dx,dy)     133.DIG  v-vector
   116 2  |33                     |      3       3 (dx,dy)     134.DIG  v-vector
   117 1  |54                     |      5       4 (dx,dy)     135.DIG  v-vector
   117 2  |32                     |      3       2 (dx,dy)     136.DIG  v-vector
   118 1  |83                     |      8       3 (dx,dy)     137.DIG  v-vector
   118 2  |A3                     |     10       3 (dx,dy)     138.DIG  v-vector
   119 1  |D3                     |     13       3 (dx,dy)     139.DIG  v-vector
   119 2  |E7 22                  |     34       7 (dX,dy)     140.DIG  s-vector
          |                       |
 ---------|-----------------------|------------------------------------------
```

```
U003003T.VE                  20-JAN-1992  11:08            PAGE  19

 ---------|------------------|--------------------------------------------------
  WORDNO. |      CONTENT     |                     COMMENT
 ---------|------------------|--------------------------------------------------
     1    |               0  | Records      length of character
     2    |             109  | Words
 ---------|------------------|--------------------------------------------------
       NAME SECTION
       ------------
     3    |               3  | Length of name section
     4    |             302  | Character number
     5    |               0  | Number of following part
 ---------|------------------|--------------------------------------------------
       SETTING INFORMATION
       -------------------
     6    |              12  | Length of setting information
     7    |               1  | Character type = letter
     8    |             116  | Number of vectors
     9    |             251  | Total set width (T=L+W+R)
    10    |              -1  | Left side bearing (L)
    11    |             234  | Width (W)
    12    |              18  | Right side bearing (R)
    13    |               0  | X-minimum
    14    |             234  | X-maximum
    15    |              -5  | Y-minimum
    16    |             276  | Y-maximum
    17    |               1  | Unit is 1 pixel
 ---------|------------------|--------------------------------------------------
       CONTOUR INDEX
       -------------
    18    |               0  | Records     length of contour index
    19    |              14  | Words
          |                  |
    20    |               0  | Record pointer for   1. contour:     0
    21    |               3  | Word pointer                   :    34
    22    |              -1  | Sense of rotation of contour = clockwise
    23    |               0  | Nesting: outer  1. order
    24    |               0  | Colour inside: transparent
    25    |              63  | Number of digitizations
          |                  |
    26    |               0  | Record pointer for   2. contour:     0
    27    |              46  | Word pointer                   :    77
    28    |              -1  | Sense of rotation of contour = clockwise
    29    |               1  | Nesting: inner  1. order
    30    |               0  | Colour inside: transparent
    31    |              53  | Number of digitizations
          |                  |
 ---------|------------------|--------------------------------------------------
       IMAGE INFORMATION
       -----------------
    32    |                    0  | Records  length of image information
    33    |                   78  | Words
          |                       |
   34-  76|      51      -5       |      51     -5 (dX,dY)      1.DIG  w-vector
   36 1  |E0 F6                   |    -10      0 (dX,dy)      2.DIG  s-vector
   37 1  |0E 80    227            |      0    227 (dx,dY)      3.DIG  l-vector
   39 1  |18                      |     -1      8 (dx,dy)      4.DIG  v-vector
   39 2  |16                      |     -1      6 (dx,dy)      5.DIG  v-vector
   40 1  |27                      |     -2      7 (dx,dy)      6.DIG  v-vector
   40 2  |34                      |     -3      4 (dx,dy)      7.DIG  v-vector
 ---------|-----------------------|---------------------------------------------
```



```
U003003T.VE                20-JAN-1992  11:09             PAGE  25
--------------------------------------------------------------------------------


                          GRAND TOTAL
                          -----------
                                             WORDS          KBYTES
                                             -----          ------
    Length of font:                          66730         133.460
    Length of font header:                    2215           4.430
    Length of all character headers:         21700          43.400
    Length of total image information:       42815          85.630


              DIGITIZATIONS                        NUMBER
              -------------                        ------
              White vectors:                         1362
              L-vectors:                             1201
              S-vectors:                            13612
              V-vectors:                            41313
                                                   ======
              Total:                                57488
              ------------------------------------------
              Characters:                             712
              Records (a 2048 words):                  33
              ------------------------------------------
              Raster (grid):                    400 X 400
              ------------------------------------------
              Digitizations per character:             81
              Image information per character:        120 bytes



--------------------------------------------------------------------------------
```

Description of the SC format

1. Structure of an SC font

The structure of an SC font is as described in Appendix G together with the extended structure as in Appendix L.

2. Description of the SC format

Pixels are grouped to form run lengths.

The scan identification (word number I + 18 in the font header) in the **scanline format** determines the direction a line progresses and where the following lines are:

The image line start is found, depending on scanning identification, at one of the sides of the square, given by XMIN, XMAX, YMIN and YMAX (words N1 + 10 up to N1 + 13). This is further standardized in that XMIN is taken to be 0 pixels and YMIN and YMAX are respective to the baseline and given in pixels (see also the sketches in Appendix L).

The lines scan (hence scanlines) across the em square, scanning a series of transitions from white to black and vice versa. The distance from one transition to the next is known as a run length. One assumes that the first run length of each line crosses a white area.

It is therefore natural to allow run length 0 to be at the start of a line for those exceptions when the line starts with a «black» run length. In the middle of the line, run lengths are always $\neq 0$. This means that run length = 0 can be, and is, used as the line end indicator, i.e. bytes 00 00 represent an empty line.

The run lengths are coded with 7, 14 or 22 bits as follows: Run length up to 127 with one byte 00...7F; run lengths up to 16,383 with 2 bytes, whereby the higher valued 7 bits are in the first byte and coded from 80 to FF. The highest value (left) bit indicates that another byte follows with 7 lower valued bits, coded as 00...7F. The two 7 bit groups form the 14 bit representation of the run length.

Still longer run lengths are coded with 22 bits and describe, in theory, lengths up to 4,194,303 points. They are indicated by the second byte with coding 80...FF. The highest value (left) bit indicates that an additional byte follows, i.e. we have then 7 + 7 + 8 bits available for coding. The first byte supplies bits 21...15, the second bits 14...8 and the third bits 7...0 for the 22 bit coding.

Coding examples

Bytes	Coded run length	Comment
00	0	Either black line start (non-white) or scanline end
00 00	0	Empty scanline
01	1	7 bit run length
0F	15	7 bit run length
7F	127	7 bit run length
81 00	128	14 bit run length
81 7F	255	14 bit run length
FF 7F	16,383	14 bit run length
80 C0 00	16,384	22 bit run length
82 80 00	65,536	22 bit run length
FF FF FF	4,194,303	22 bit run length

The number of line repetitions is written just behind the run lengths of a scanline, if applicable. As a scanline always has to have an even number of run lengths, this notation gives an odd number and can be easily recognized.

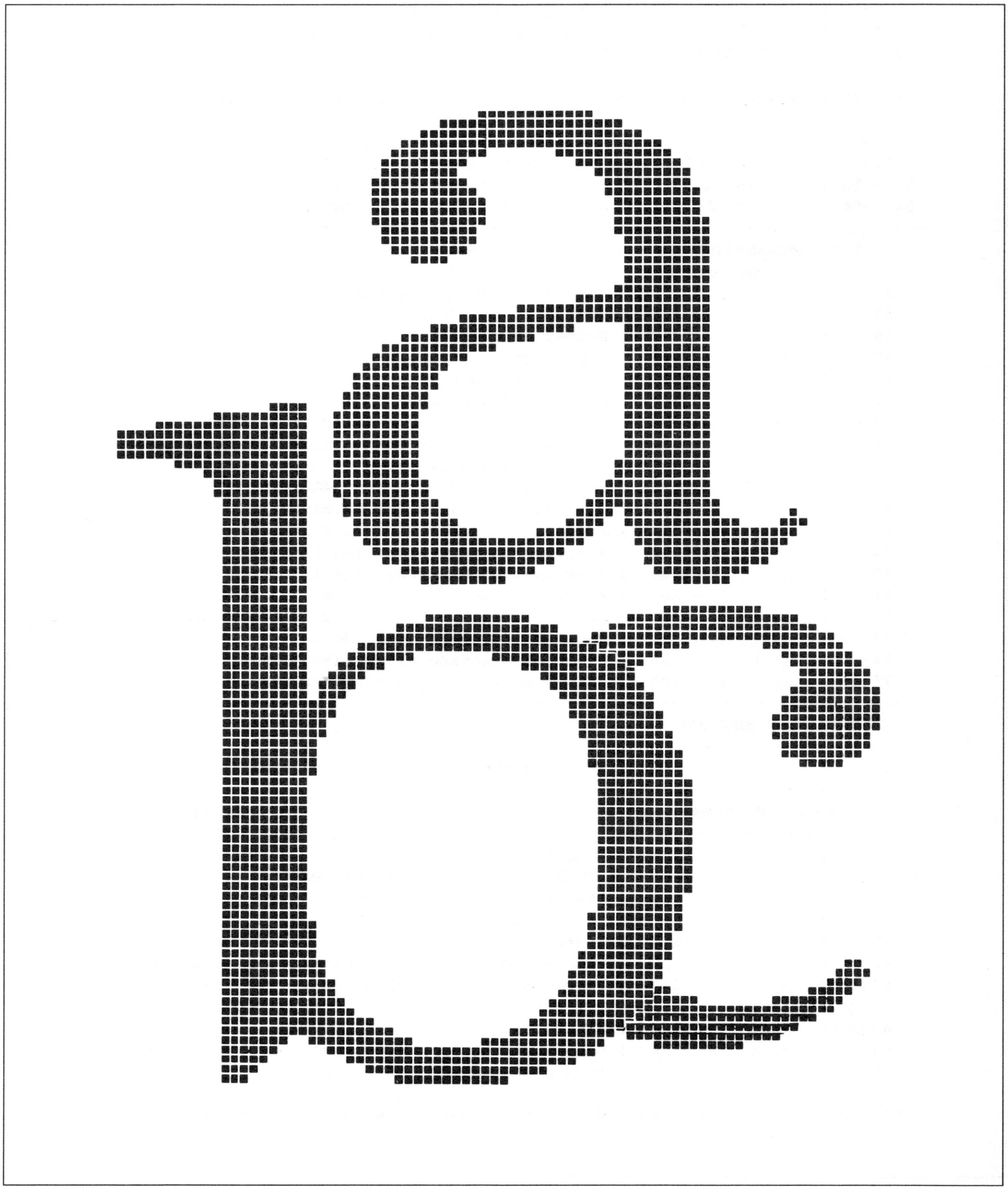
The end of the character is controlled using the line index and correspondingly, the end of a font is checked by the character index.

One run length, no matter whether coded with 7, 14 or 22 bits, is counted as one digitization.

3. Example of an SC format

First are represented the lower case letters «a», «b», «c» of the typeface «URW Antiqua 2015 normal», as a raster image. The scanning identification is 21 (CRT machines). The raster is 100 × 100 for the em. Subsequently the type font format is reproduced as a data list.

Fig. 197
SC format representation

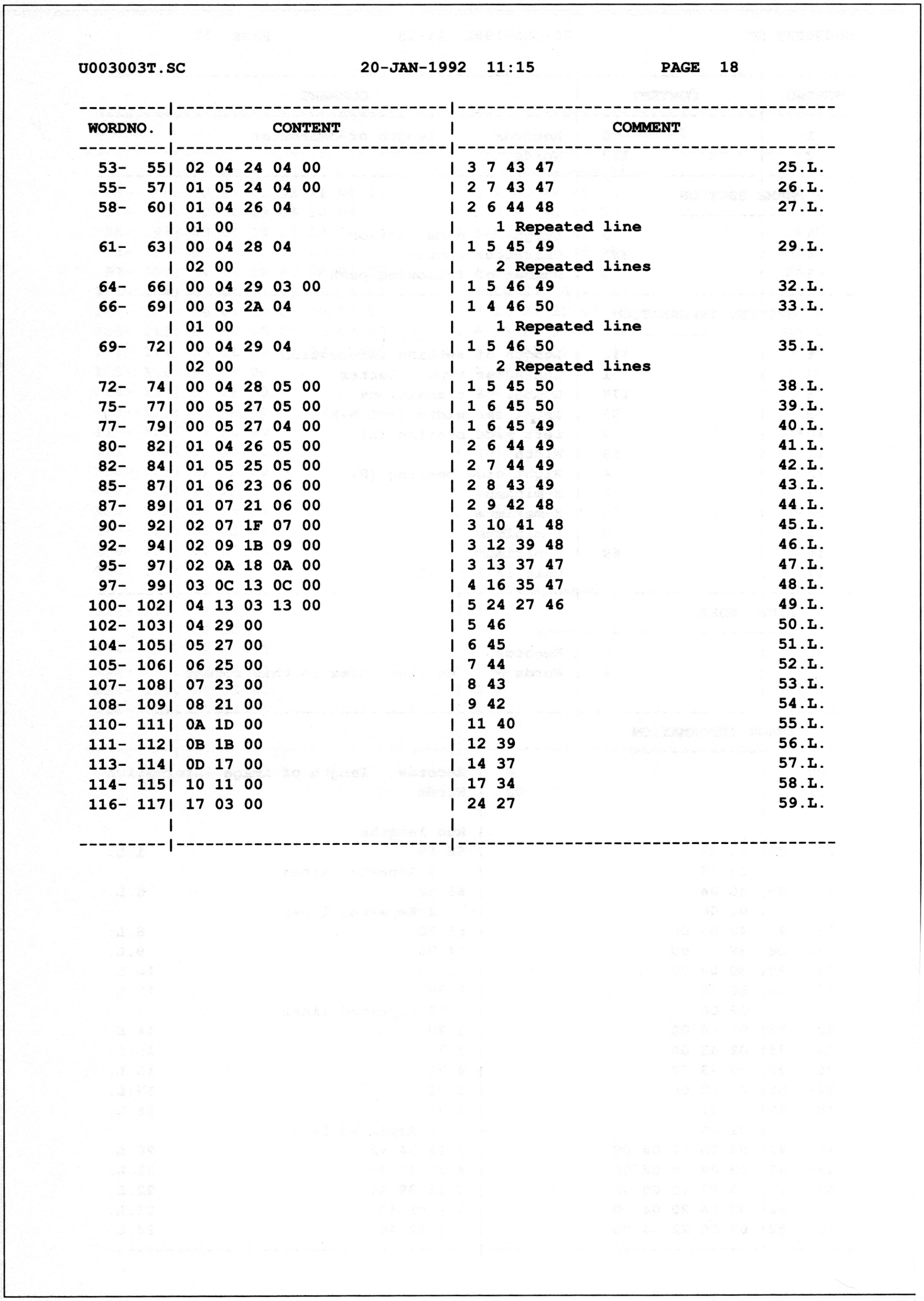


```
U003003T.SC                  20-JAN-1992  11:15            PAGE   1

---------|------------------|------------------------------------------------
 WORDNO. |     CONTENT      |                  COMMENT
---------|------------------|------------------------------------------------
    1    |             2215 | Length of font header
---------|------------------|------------------------------------------------
      NAME SECTION
      ------------

    2    |               55 | Length of name section
    3    |                0 | Number on URW list
  4 - 10 |  U003003T.SC     | File name (ASCII)
         |                  |
 11 - 49 |URW Antiqua 2015 Regular              | Font name (ASCII)
         |
         |                  |
   50    |               SC | Data format (ASCII)
 51 - 53 |      20   1 1992 | Production date:     20-Jan-1992
 54 - 56 |      20   1 1992 | Date of last change: 20-Jan-1992
---------|------------------|------------------------------------------------
      FONT INFORMATION
      ----------------

   57    |               19 | Length of font information
   58    |                1 | Indicator for font
   59    |              712 | Number of characters
   60    |               67 | Cap height (in pixels)
   61    |              100 | Body size (in pixels)
   62    |               47 | x-height (in pixels)
   63    |               22 | Distance: base line - lower body line(in pixels)
   64    |              100 | Text line distance for setting (in pixels)
   65    |                9 | Stem thickness (in pixels)
   66    |                0 | Angle of italisation (in 1/10 degree)
   67    |               12 | Optimum point size for setting (in p)
   68    |             8462 | Average character width (in pixels)
   69    |             1500 | X-extension of a pixel (in 1/1000mm)
   70    |             1500 | Y-extension of a pixel (in 1/1000mm)
   71    |              100 | Em-resolution in X-direction (in pixels)
   72    |              100 | Em-resolution in Y-direction (in pixels)
   73    |               21 | Scan indicator of image data
   74    |              100 | Width of printing window (in pixels)
   75    |              100 | Height of printing window (in pixels)
---------|------------------|------------------------------------------------
      HIERARCHY SECTION
      -----------------

   76    |                1 | No hierarchy section in this format
---------|------------------|------------------------------------------------
       CHARACTER INDEX
      ----------------

   77    |             2139 | Length of character index
   78    |               39 | LENGTH OF FONT IN PHYSICAL RECORDS
   79    |             1011 | Last word of last record
         |                  |
   80    |              101 | Character number:  1
   81    |                2 | Pointer to record containing start of character
   82    |              168 | Word pointer to start of character
         |                  |
   83    |              102 | Ch. no.: 2
   84    |                2 | Record pointer
   85    |              334 | Word pointer
         |                  |
  86-  88|   103     2  451 | No., record pointer, word pointer of ch.:    3
---------|------------------|------------------------------------------------
```

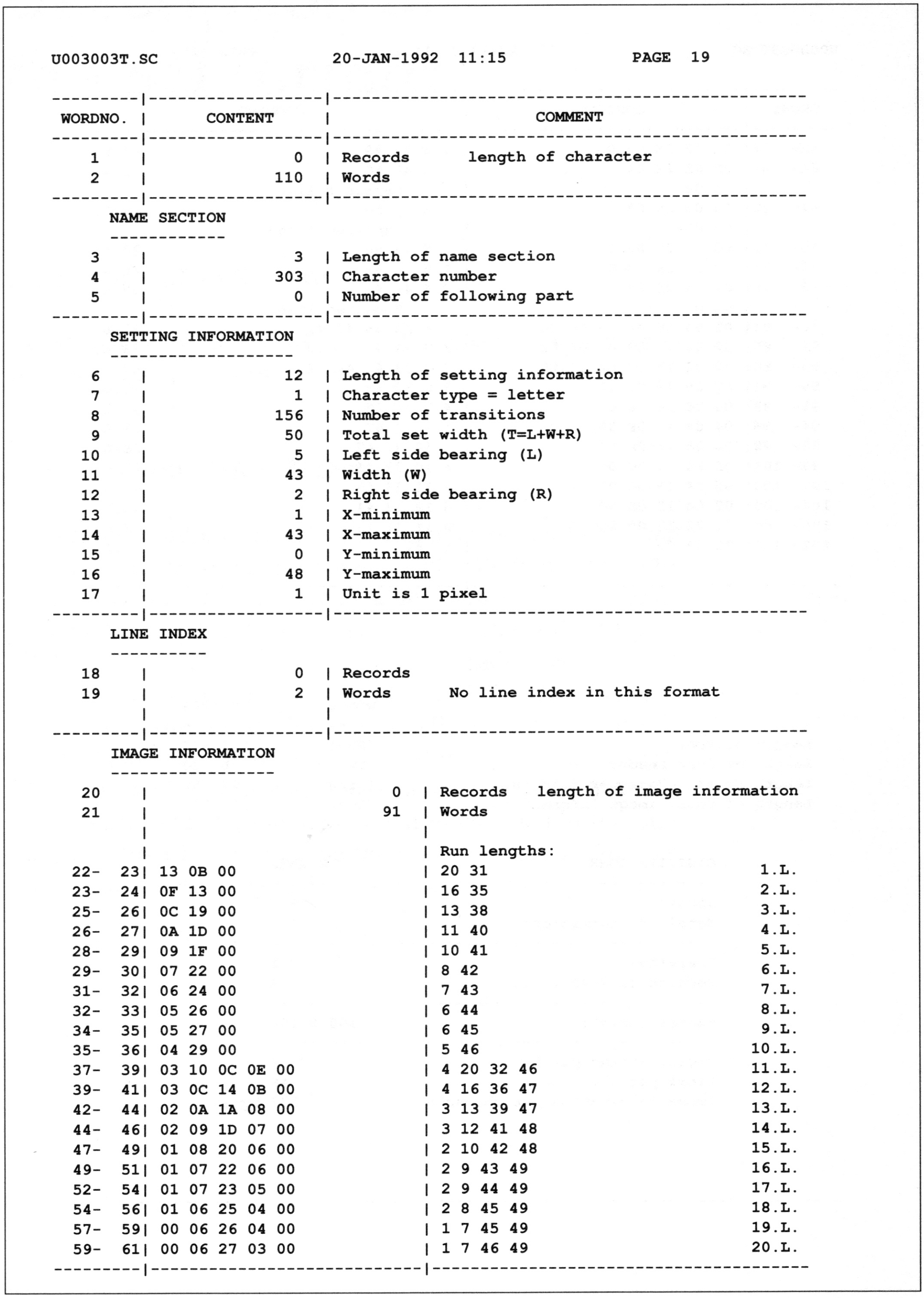
```
U003003T.SC                   20-JAN-1992  11:15            PAGE   2

---------|------------------|---------------------------------------------
 WORDNO. |      CONTENT     |                  COMMENT
---------|------------------|---------------------------------------------
  89-  91|  104      2  590 | No., record pointer, word pointer of ch.:    4
  92-  94|  105      2  701 | No., record pointer, word pointer of ch.:    5
  95-  97|  106      2  816 | No., record pointer, word pointer of ch.:    6
  98- 100|  107      2  908 | No., record pointer, word pointer of ch.:    7
 101- 103|  108      2 1070 | No., record pointer, word pointer of ch.:    8
 104- 106|  109      2 1169 | No., record pointer, word pointer of ch.:    9
 107- 109|  110      2 1222 | No., record pointer, word pointer of ch.:   10
 110- 112|  111      2 1290 | No., record pointer, word pointer of ch.:   11
 113- 115|  112      2 1454 | No., record pointer, word pointer of ch.:   12
 116- 118|  113      2 1525 | No., record pointer, word pointer of ch.:   13
 119- 121|  114      2 1691 | No., record pointer, word pointer of ch.:   14
 122- 124|  115      2 1833 | No., record pointer, word pointer of ch.:   15
 125- 127|  116      2 1987 | No., record pointer, word pointer of ch.:   16
 128- 130|  117      3   46 | No., record pointer, word pointer of ch.:   17
 131- 133|  118      3  262 | No., record pointer, word pointer of ch.:   18
 134- 136|  119      3  389 | No., record pointer, word pointer of ch.:   19
 137- 139|  120      3  546 | No., record pointer, word pointer of ch.:   20
 140- 142|  121      3  635 | No., record pointer, word pointer of ch.:   21




 404- 406|  285     10 1135 | No., record pointer, word pointer of ch.:  109
 407- 409|  286     10 1302 | No., record pointer, word pointer of ch.:  110
 410- 412|  287     10 1501 | No., record pointer, word pointer of ch.:  111
 413- 415|  289     10 1697 | No., record pointer, word pointer of ch.:  112
 416- 418|  290     10 1866 | No., record pointer, word pointer of ch.:  113
 419- 421|  291     10 1996 | No., record pointer, word pointer of ch.:  114
 422- 424|  292     11   81 | No., record pointer, word pointer of ch.:  115
 425- 427|  296     11  259 | No., record pointer, word pointer of ch.:  116
 428- 430|  297     11  416 | No., record pointer, word pointer of ch.:  117
 431- 433|  298     11  527 | No., record pointer, word pointer of ch.:  118
 434- 436|  301     11  739 | No., record pointer, word pointer of ch.:  119
 437- 439|  302     11  878 | No., record pointer, word pointer of ch.:  120
 440- 442|  303     11  995 | No., record pointer, word pointer of ch.:  121




2171-2173| 2121     38 1369 | No., record pointer, word pointer of ch.:  698
2174-2176| 2124     38 1544 | No., record pointer, word pointer of ch.:  699
2177-2179| 2301     38 1752 | No., record pointer, word pointer of ch.:  700
2180-2182| 2302     38 1882 | No., record pointer, word pointer of ch.:  701
2183-2185| 2304     38 2004 | No., record pointer, word pointer of ch.:  702
2186-2188| 2305     39  112 | No., record pointer, word pointer of ch.:  703
2189-2191| 2307     39  224 | No., record pointer, word pointer of ch.:  704
2192-2194| 2312     39  292 | No., record pointer, word pointer of ch.:  705
2195-2197| 2316     39  370 | No., record pointer, word pointer of ch.:  706
2198-2200| 2318     39  470 | No., record pointer, word pointer of ch.:  707
2201-2203| 2319     39  583 | No., record pointer, word pointer of ch.:  708
2204-2206| 2321     39  657 | No., record pointer, word pointer of ch.:  709
2207-2209| 2323     39  788 | No., record pointer, word pointer of ch.:  710
2210-2212| 2325     39  882 | No., record pointer, word pointer of ch.:  711
2213-2215| 20392    39  991 | No., record pointer, word pointer of ch.:  712
---------|------------------|---------------------------------------------
```

```
U003003T.SC                    20-JAN-1992  11:15            PAGE  16

---------|----------------------------|---------------------------------------
 WORDNO. |           CONTENT          |                 COMMENT
---------|----------------------------|---------------------------------------
         | 01 00                      |    1 Repeated line
  83-  86| 02 04 13 03 11 04 00       | 3 7 26 29 46 50                  22.L.
  87-  90| 02 04 13 04 10 04 00       | 3 7 26 30 46 50                  23.L.
  90-  93| 03 04 13 03 10 04 00       | 4 8 27 30 46 50                  24.L.
  94-  97| 04 03 13 03 10 04 00       | 5 8 27 30 46 50                  25.L.
  97- 100| 04 04 12 03 0F 04 00       | 5 9 27 30 45 49                  26.L.
 101- 104| 05 04 11 03 0F 04 00       | 6 10 27 30 45 49                 27.L.
 104- 107| 06 04 10 04 0D 05 00       | 7 11 27 31 44 49                 28.L.
 108- 111| 07 04 10 03 0B 07 00       | 8 12 28 31 42 49                 29.L.
 111- 114| 08 06 0D 03 08 0A 00       | 9 15 28 31 39 49                 30.L.
 115- 116| 06 29 00                   | 7 48                             31.L.
 116- 117| 04 2B 00                   | 5 48                             32.L.
 118- 119| 03 2B 00                   | 4 47                             33.L.
 119- 120| 02 2C 00                   | 3 47                             34.L.
 121- 122| 01 2C 00                   | 2 46                             35.L.
 122- 123| 01 2B 00                   | 2 45                             36.L.
 124- 125| 00 2A 00                   | 1 43                             37.L.
 125- 126| 00 29 00                   | 1 42                             38.L.
 127- 128| 00 25 00                   | 1 38                             39.L.
 128- 129| 01 07 00                   | 2 9                              40.L.
 130- 131| 01 05                      | 2 7                              41.L.
         | 01 00                      |    1 Repeated line
 132- 133| 02 04                      | 3 7                              43.L.
         | 01 00                      |    1 Repeated line
 134- 135| 03 03 00                   | 4 7                              45.L.
 135- 136| 03 04 00                   | 4 8                              46.L.
 137- 138| 04 03 00                   | 5 8                              47.L.
 138- 139| 05 03 00                   | 6 9                              48.L.
         |                            |
---------|----------------------------|---------------------------------------
```

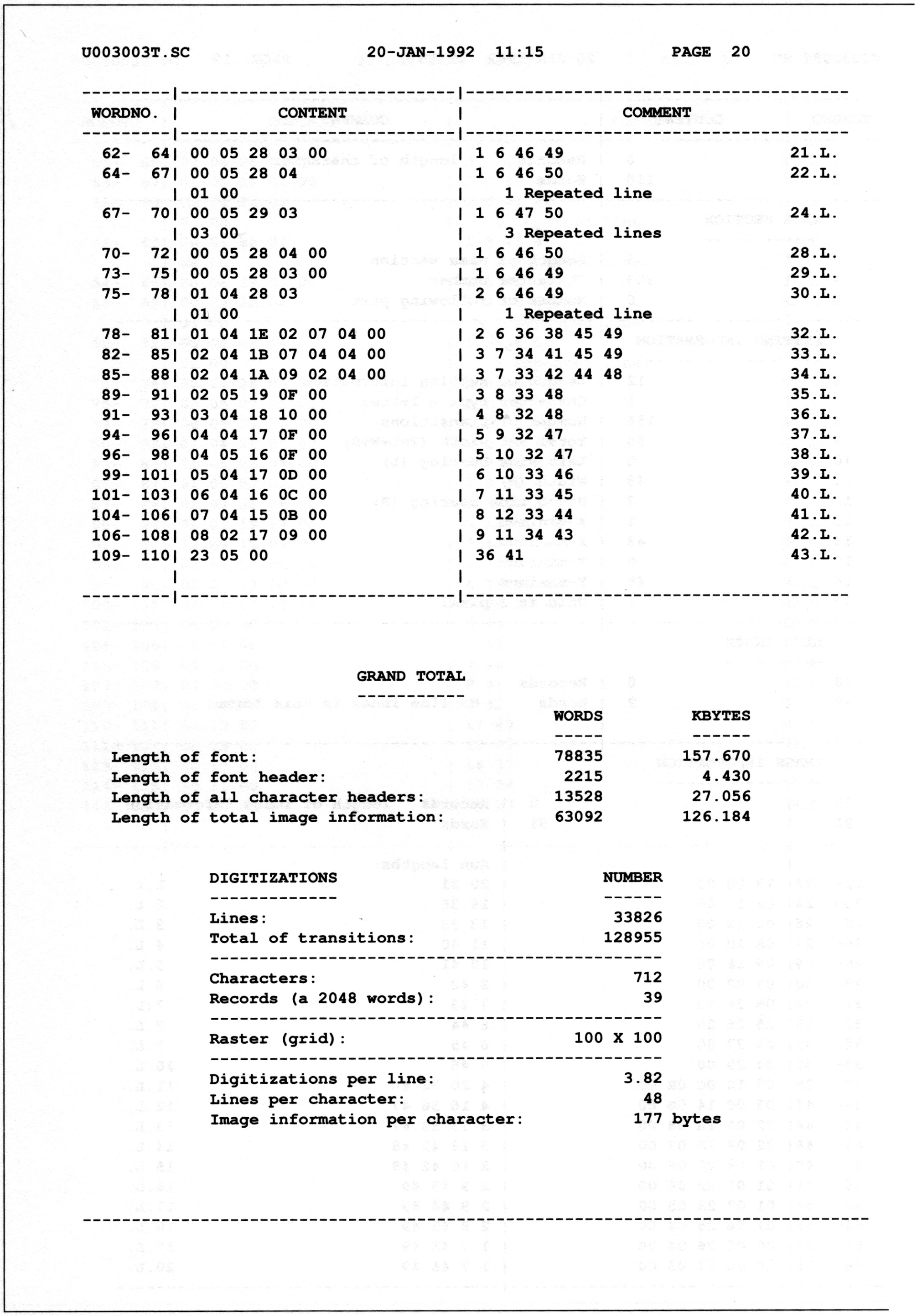
```
U003003T.SC                 20-JAN-1992  11:15           PAGE  17

---------|------------------|---------------------------------------------
 WORDNO. |      CONTENT     |                   COMMENT
---------|------------------|---------------------------------------------
    1    |                0 | Records      length of character
    2    |              117 | Words
---------|------------------|---------------------------------------------
      NAME SECTION
      ------------
    3    |                3 | Length of name section
    4    |              302 | Character number
    5    |                0 | Number of following part
---------|------------------|---------------------------------------------
      SETTING INFORMATION
      -------------------
    6    |               12 | Length of setting information
    7    |                1 | Character type = letter
    8    |              178 | Number of transitions
    9    |               63 | Total set width (T=L+W+R)
   10    |                0 | Left side bearing (L)
   11    |               59 | Width (W)
   12    |                4 | Right side bearing (R)
   13    |                1 | X-minimum
   14    |               59 | X-maximum
   15    |                0 | Y-minimum
   16    |               69 | Y-maximum
   17    |                1 | Unit is 1 pixel
---------|------------------|---------------------------------------------
      LINE INDEX
      ----------
   18    |                0 | Records
   19    |                2 | Words      No line index in this format
         |                  |
---------|------------------|---------------------------------------------
      IMAGE INFORMATION
      -----------------
   20    |                          0  | Records   length of image information
   21    |                         98  | Words
         |                             |
         |                             | Run lengths:
  22-  23| 41 03                       | 66 69                           1.L.
         | 03 00                       |   3 Repeated lines
  24-  25| 40 04                       | 65 69                           5.L.
         | 02 00                       |   2 Repeated lines
  26-  27| 40 05 00                    | 65 70                           8.L.
  27-  28| 3F 06 00                    | 64 70                           9.L.
  29-  30| 3D 08 00                    | 62 70                          10.L.
  30-  32| 00 45                       | 1 70                           11.L.
         | 02 00                       |   2 Repeated lines
  32-  33| 01 44 00                    | 2 70                           14.L.
  34-  35| 02 43 00                    | 3 70                           15.L.
  35-  36| 03 43 00                    | 4 71                           16.L.
  37-  38| 04 42 00                    | 5 71                           17.L.
  38-  40| 05 41                       | 6 71                           18.L.
         | 01 00                       |   1 Repeated line
  40-  42| 04 0B 12 08 00              | 5 16 34 42                     20.L.
  43-  45| 03 09 18 06 00              | 4 13 37 43                     21.L.
  45-  47| 03 07 1C 05 00              | 4 11 39 44                     22.L.
  48-  50| 02 06 20 04 00              | 3 9 41 45                      23.L.
  50-  52| 02 05 22 04 00              | 3 8 42 46                      24.L.
---------|-----------------------------|-----------------------------------
```



```
U003003T.SC                20-JAN-1992  11:15            PAGE  18

---------|---------------------------|---------------------------------------
 WORDNO. |          CONTENT          |                COMMENT
---------|---------------------------|---------------------------------------
  53-  55| 02 04 24 04 00            | 3 7 43 47                        25.L.
  55-  57| 01 05 24 04 00            | 2 7 43 47                        26.L.
  58-  60| 01 04 26 04               | 2 6 44 48                        27.L.
         | 01 00                     |    1 Repeated line
  61-  63| 00 04 28 04               | 1 5 45 49                        29.L.
         | 02 00                     |    2 Repeated lines
  64-  66| 00 04 29 03 00            | 1 5 46 49                        32.L.
  66-  69| 00 03 2A 04               | 1 4 46 50                        33.L.
         | 01 00                     |    1 Repeated line
  69-  72| 00 04 29 04               | 1 5 46 50                        35.L.
         | 02 00                     |    2 Repeated lines
  72-  74| 00 04 28 05 00            | 1 5 45 50                        38.L.
  75-  77| 00 05 27 05 00            | 1 6 45 50                        39.L.
  77-  79| 00 05 27 04 00            | 1 6 45 49                        40.L.
  80-  82| 01 04 26 05 00            | 2 6 44 49                        41.L.
  82-  84| 01 05 25 05 00            | 2 7 44 49                        42.L.
  85-  87| 01 06 23 06 00            | 2 8 43 49                        43.L.
  87-  89| 01 07 21 06 00            | 2 9 42 48                        44.L.
  90-  92| 02 07 1F 07 00            | 3 10 41 48                       45.L.
  92-  94| 02 09 1B 09 00            | 3 12 39 48                       46.L.
  95-  97| 02 0A 18 0A 00            | 3 13 37 47                       47.L.
  97-  99| 03 0C 13 0C 00            | 4 16 35 47                       48.L.
 100- 102| 04 13 03 13 00            | 5 24 27 46                       49.L.
 102- 103| 04 29 00                  | 5 46                             50.L.
 104- 105| 05 27 00                  | 6 45                             51.L.
 105- 106| 06 25 00                  | 7 44                             52.L.
 107- 108| 07 23 00                  | 8 43                             53.L.
 108- 109| 08 21 00                  | 9 42                             54.L.
 110- 111| 0A 1D 00                  | 11 40                            55.L.
 111- 112| 0B 1B 00                  | 12 39                            56.L.
 113- 114| 0D 17 00                  | 14 37                            57.L.
 114- 115| 10 11 00                  | 17 34                            58.L.
 116- 117| 17 03 00                  | 24 27                            59.L.
         |                           |
---------|---------------------------|---------------------------------------
```



```
U003003T.SC                  20-JAN-1992  11:15            PAGE  19

---------|------------------|------------------------------------------------
 WORDNO. |      CONTENT     |                    COMMENT
---------|------------------|------------------------------------------------
    1    |                0 | Records     length of character
    2    |              110 | Words
---------|------------------|------------------------------------------------
      NAME SECTION
      ------------
    3    |                3 | Length of name section
    4    |              303 | Character number
    5    |                0 | Number of following part
---------|------------------|------------------------------------------------
      SETTING INFORMATION
      -------------------
    6    |               12 | Length of setting information
    7    |                1 | Character type = letter
    8    |              156 | Number of transitions
    9    |               50 | Total set width (T=L+W+R)
   10    |                5 | Left side bearing (L)
   11    |               43 | Width (W)
   12    |                2 | Right side bearing (R)
   13    |                1 | X-minimum
   14    |               43 | X-maximum
   15    |                0 | Y-minimum
   16    |               48 | Y-maximum
   17    |                1 | Unit is 1 pixel
---------|------------------|------------------------------------------------
      LINE INDEX
      ----------
   18    |                0 | Records
   19    |                2 | Words      No line index in this format
         |                  |
---------|------------------|------------------------------------------------
      IMAGE INFORMATION
      -----------------
   20    |                           0 | Records   length of image information
   21    |                          91 | Words
         |                             |
         |                             | Run lengths:
  22-  23| 13 0B 00                    | 20 31                          1.L.
  23-  24| 0F 13 00                    | 16 35                          2.L.
  25-  26| 0C 19 00                    | 13 38                          3.L.
  26-  27| 0A 1D 00                    | 11 40                          4.L.
  28-  29| 09 1F 00                    | 10 41                          5.L.
  29-  30| 07 22 00                    | 8 42                           6.L.
  31-  32| 06 24 00                    | 7 43                           7.L.
  32-  33| 05 26 00                    | 6 44                           8.L.
  34-  35| 05 27 00                    | 6 45                           9.L.
  35-  36| 04 29 00                    | 5 46                          10.L.
  37-  39| 03 10 0C 0E 00              | 4 20 32 46                    11.L.
  39-  41| 03 0C 14 0B 00              | 4 16 36 47                    12.L.
  42-  44| 02 0A 1A 08 00              | 3 13 39 47                    13.L.
  44-  46| 02 09 1D 07 00              | 3 12 41 48                    14.L.
  47-  49| 01 08 20 06 00              | 2 10 42 48                    15.L.
  49-  51| 01 07 22 06 00              | 2 9 43 49                     16.L.
  52-  54| 01 07 23 05 00              | 2 9 44 49                     17.L.
  54-  56| 01 06 25 04 00              | 2 8 45 49                     18.L.
  57-  59| 00 06 26 04 00              | 1 7 45 49                     19.L.
  59-  61| 00 06 27 03 00              | 1 7 46 49                     20.L.
---------|-----------------------------|-------------------------------------
```



```
U003003T.SC                20-JAN-1992  11:15            PAGE  20

---------|----------------------------|--------------------------------------
WORDNO. |           CONTENT           |                COMMENT
---------|----------------------------|--------------------------------------
  62-  64| 00 05 28 03 00             | 1 6 46 49                        21.L.
  64-  67| 00 05 28 04                | 1 6 46 50                        22.L.
         | 01 00                      |     1 Repeated line
  67-  70| 00 05 29 03                | 1 6 47 50                        24.L.
         | 03 00                      |     3 Repeated lines
  70-  72| 00 05 28 04 00             | 1 6 46 50                        28.L.
  73-  75| 00 05 28 03 00             | 1 6 46 49                        29.L.
  75-  78| 01 04 28 03                | 2 6 46 49                        30.L.
         | 01 00                      |     1 Repeated line
  78-  81| 01 04 1E 02 07 04 00       | 2 6 36 38 45 49                  32.L.
  82-  85| 02 04 1B 07 04 04 00       | 3 7 34 41 45 49                  33.L.
  85-  88| 02 04 1A 09 02 04 00       | 3 7 33 42 44 48                  34.L.
  89-  91| 02 05 19 0F 00             | 3 8 33 48                        35.L.
  91-  93| 03 04 18 10 00             | 4 8 32 48                        36.L.
  94-  96| 04 04 17 0F 00             | 5 9 32 47                        37.L.
  96-  98| 04 05 16 0F 00             | 5 10 32 47                       38.L.
  99- 101| 05 04 17 0D 00             | 6 10 33 46                       39.L.
 101- 103| 06 04 16 0C 00             | 7 11 33 45                       40.L.
 104- 106| 07 04 15 0B 00             | 8 12 33 44                       41.L.
 106- 108| 08 02 17 09 00             | 9 11 34 43                       42.L.
 109- 110| 23 05 00                   | 36 41                            43.L.
         |                            |
---------|----------------------------|--------------------------------------


                         GRAND TOTAL
                         -----------
                                              WORDS          KBYTES
                                              -----          ------
  Length of font:                             78835         157.670
  Length of font header:                       2215           4.430
  Length of all character headers:            13528          27.056
  Length of total image information:          63092         126.184


         DIGITIZATIONS                              NUMBER
         -------------                              ------
         Lines:                                      33826
         Total of transitions:                      128955
         -------------------------------------------------
         Characters:                                   712
         Records (a 2048 words):                        39
         -------------------------------------------------
         Raster (grid):                          100 X 100
         -------------------------------------------------
         Digitizations per line:                      3.82
         Lines per character:                           48
         Image information per character:          177 bytes



-----------------------------------------------------------------------------
```

Description of the SN format

1. Structure of an SN font

The structure of an SN font is as described in Appendix G together with the extended structure as in Appendix L.

2. Description of the SN format

The **scanline nibble format** is a variation of the SC format especially suited for rasters between 30 × 30 up to 140 × 140. We propose to use this format rather than bitmaps in laser printers. SN requires less storage capacity (see grand total of SC and SN format).

Basically, a byte is replaced by a halfbyte, the nibble, so one may describe a very short run length by a nibble. The nibbles 8...F indicate that the high value three bits are given with this nibble, and the low value three bits follow in the next nibble. One uses as many nibbles as are needed. Each further nibble gives three bits more. The later a nibble occurs in a sequence for one run length, the lower its contributing value.

Coding examples

Nibbles	Coded run length	Comment
0	0	Either black line start (non-white) or scanline end
00	0	Empty scanline
1	1	3 bit run length
7	7	3 bit run length
90	8	6 bit run length
F7	63	6 bit run length
980	64	9 bit run length
FF7	511	9 bit run length
9880	512	12 bit run length
FFF7	4095	12 bit run length
98880	4096	15 bit run length
.	.	.
.	.	.
.	.	.
etc.	etc.	etc.

The number of line repetitions is written just behind the run lengths of a scanline, if applicable. As a scanline must always have an even number of run lengths, this notation gives an odd number and can be easily recognized.

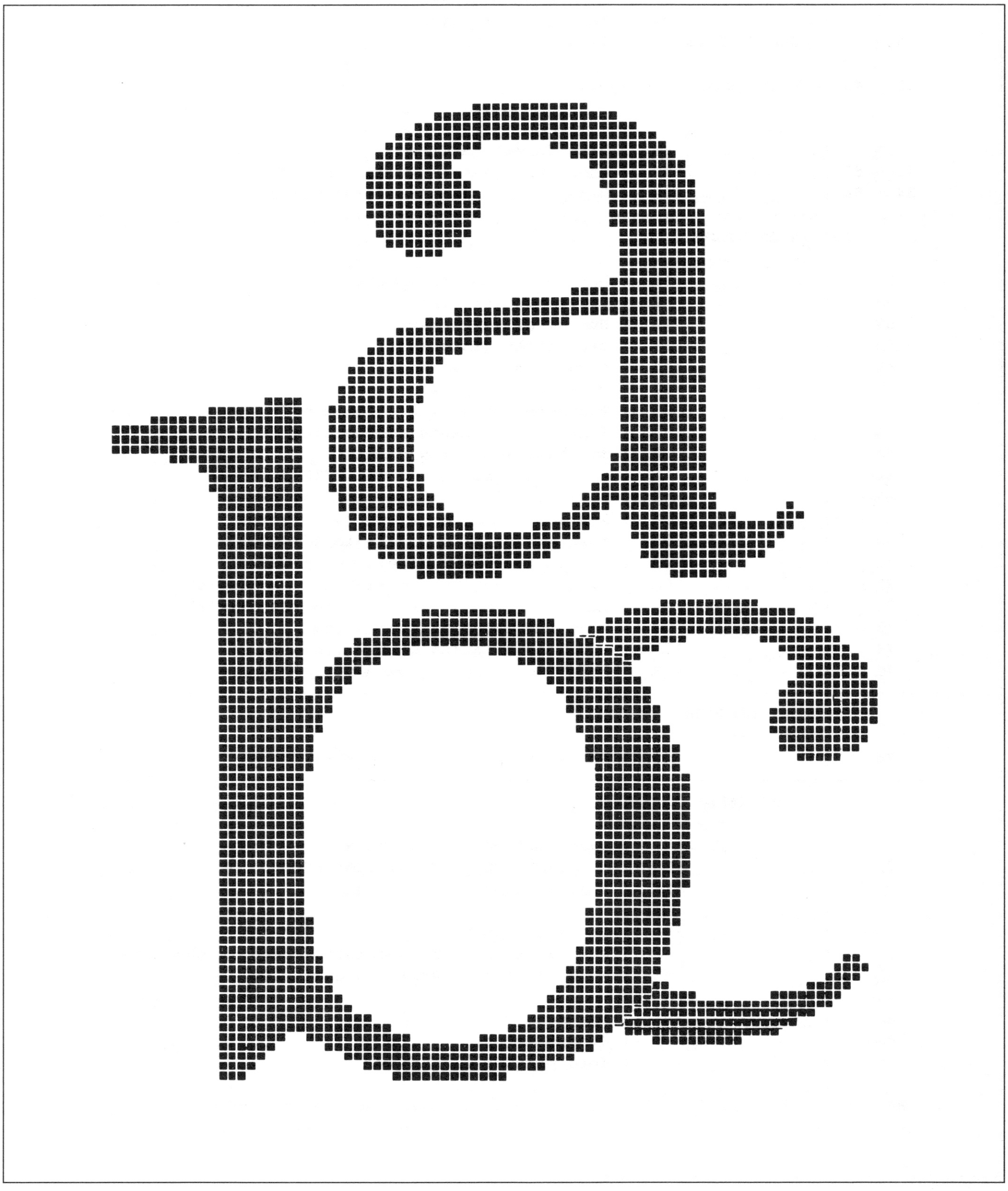
The end of the character is controlled using the line index and correspondingly, the end of a font is checked by the character index.

One run length, no matter whether coded with 3, 6, 9, 12 or 15 bits, is counted as one digitization.

3. Example of an SN format

First are represented the lower case letters «a», «b», «c» of the typeface «URW Antiqua 2015 normal», as a raster image. The scanning identification is 21 (CRT machines). The raster is 100 × 100 for the em. Subsequently the type font format is reproduced as a data list.

Fig. 198
SN format representation



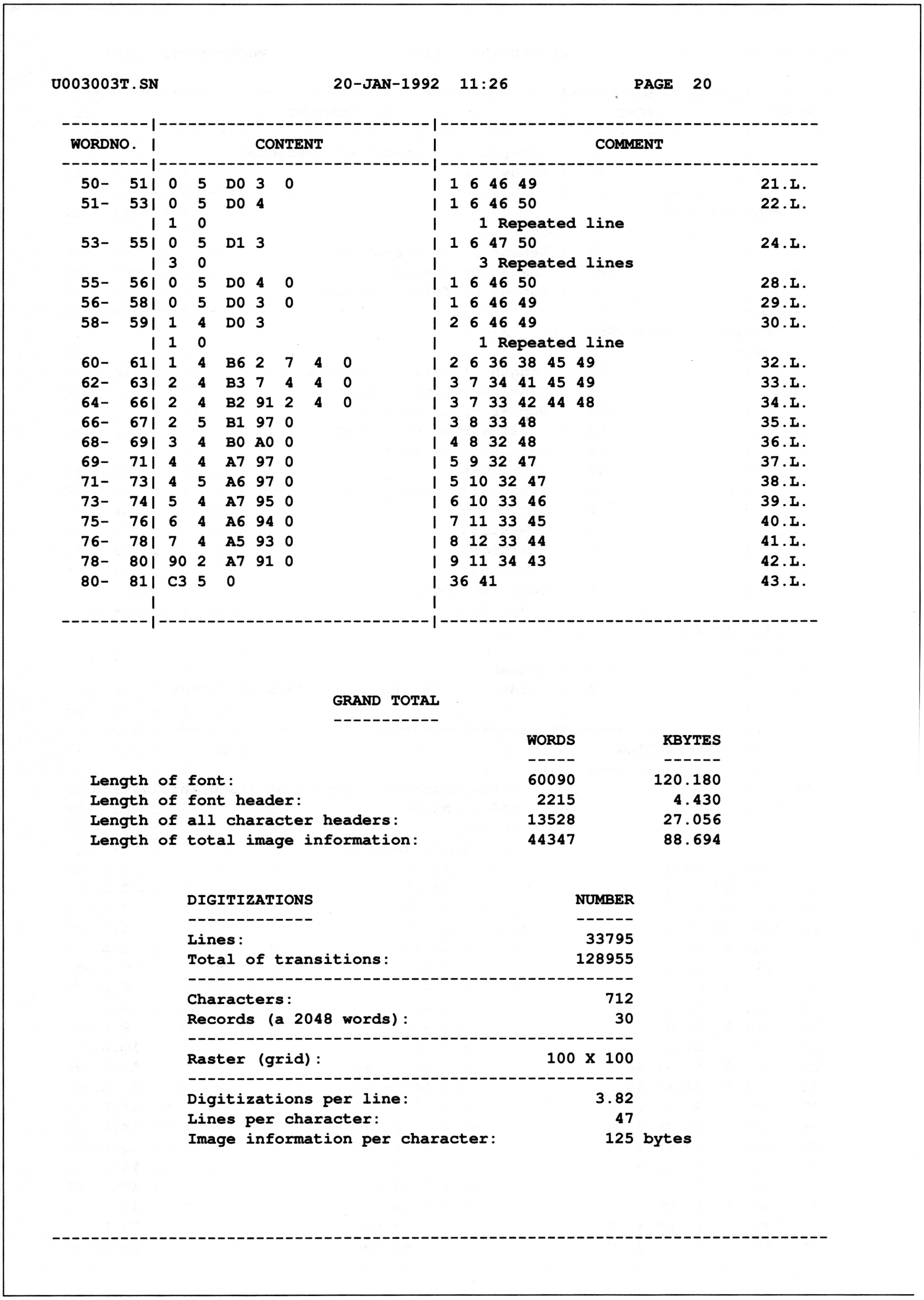
```
     U003003T.SN                20-JAN-1992  11:25             PAGE   1

  ---------|------------------|-----------------------------------------------
   WORDNO. |     CONTENT      |                COMMENT
  ---------|------------------|-----------------------------------------------
      1    |           2215   | Length of font header
  ---------|------------------|-----------------------------------------------
        NAME SECTION
        ------------

      2    |              55  | Length of name section
      3    |               0  | Number on URW list
    4 - 10 |  U003003T.SN     | File name (ASCII)
           |                  |
   11 - 49 |URW Antiqua 2015 Regular              | Font name (ASCII)
           |
           |                  |
     50    |              SN  | Data format (ASCII)
   51 - 53 |      20  1 1992  | Production date:     20-Jan-1992
   54 - 56 |      20  1 1992  | Date of last change: 20-Jan-1992
  ---------|------------------|-----------------------------------------------
        FONT INFORMATION
        ----------------

     57    |              19  | Length of font information
     58    |               1  | Indicator for font
     59    |             712  | Number of characters
     60    |              67  | Cap height (in pixels)
     61    |             100  | Body size (in pixels)
     62    |              47  | x-height (in pixels)
     63    |              22  | Distance: base line - lower body line(in pixels)
     64    |             100  | Text line distance for setting (in pixels)
     65    |               9  | Stem thickness (in pixels)
     66    |               0  | Angle of italisation (in 1/10 degree)
     67    |              12  | Optimum point size for setting (in p)
     68    |            8462  | Average character width (in pixels)
     69    |            1500  | X-extension of a pixel (in 1/1000mm)
     70    |            1500  | Y-extension of a pixel (in 1/1000mm)
     71    |             100  | Em-resolution in X-direction (in pixels)
     72    |             100  | Em-resolution in Y-direction (in pixels)
     73    |              21  | Scan indicator of image data
     74    |             100  | Width of printing window (in pixels)
     75    |             100  | Height of printing window (in pixels)
  ---------|------------------|-----------------------------------------------
        HIERARCHY SECTION
        -----------------
     76    |               1  | No hierarchy section in this format
  ---------|------------------|-----------------------------------------------
         CHARACTER INDEX
         ---------------
     77    |            2139  | Length of character index
     78    |              30  | LENGTH OF FONT IN PHYSICAL RECORDS
     79    |             698  | Last word of last record
           |                  |
     80    |             101  | Character number:  1
     81    |               2  | Pointer to record containing start of character
     82    |             168  | Word pointer to start of character
           |                  |
     83    |             102  | Ch. no.: 2
     84    |               2  | Record pointer
     85    |             290  | Word pointer
           |                  |
    86-  88|   103     2  380 | No., record pointer, word pointer of ch.:     3
  ---------|------------------|-----------------------------------------------
```

```
U003003T.SN                  20-JAN-1992  11:25            PAGE   2

 ---------|------------------|-------------------------------------------------
  WORDNO. |     CONTENT      |                   COMMENT
 ---------|------------------|-------------------------------------------------
   89-  91|   104     2  484 | No., record pointer, word pointer of ch.:    4
   92-  94|   105     2  567 | No., record pointer, word pointer of ch.:    5
   95-  97|   106     2  653 | No., record pointer, word pointer of ch.:    6
   98- 100|   107     2  723 | No., record pointer, word pointer of ch.:    7
  101- 103|   108     2  843 | No., record pointer, word pointer of ch.:    8
  104- 106|   109     2  915 | No., record pointer, word pointer of ch.:    9
  107- 109|   110     2  956 | No., record pointer, word pointer of ch.:   10
  110- 112|   111     2 1013 | No., record pointer, word pointer of ch.:   11
  113- 115|   112     2 1134 | No., record pointer, word pointer of ch.:   12
  116- 118|   113     2 1185 | No., record pointer, word pointer of ch.:   13
  119- 121|   114     2 1309 | No., record pointer, word pointer of ch.:   14
  122- 124|   115     2 1419 | No., record pointer, word pointer of ch.:   15
  125- 127|   116     2 1536 | No., record pointer, word pointer of ch.:   16
  128- 130|   117     2 1619 | No., record pointer, word pointer of ch.:   17
  131- 133|   118     2 1790 | No., record pointer, word pointer of ch.:   18
  134- 136|   119     2 1886 | No., record pointer, word pointer of ch.:   19
  137- 139|   120     2 2009 | No., record pointer, word pointer of ch.:   20
  140- 142|   121     3   29 | No., record pointer, word pointer of ch.:   21




  404- 406|   285     8  831 | No., record pointer, word pointer of ch.:  109
  407- 409|   286     8  960 | No., record pointer, word pointer of ch.:  110
  410- 412|   287     8 1111 | No., record pointer, word pointer of ch.:  111
  413- 415|   289     8 1262 | No., record pointer, word pointer of ch.:  112
  416- 418|   290     8 1388 | No., record pointer, word pointer of ch.:  113
  419- 421|   291     8 1486 | No., record pointer, word pointer of ch.:  114
  422- 424|   292     8 1588 | No., record pointer, word pointer of ch.:  115
  425- 427|   296     8 1719 | No., record pointer, word pointer of ch.:  116
  428- 430|   297     8 1835 | No., record pointer, word pointer of ch.:  117
  431- 433|   298     8 1919 | No., record pointer, word pointer of ch.:  118
  434- 436|   301     9   23 | No., record pointer, word pointer of ch.:  119
  437- 439|   302     9  121 | No., record pointer, word pointer of ch.:  120
  440- 442|   303     9  208 | No., record pointer, word pointer of ch.:  121




 2171-2173|  2121    29 1455 | No., record pointer, word pointer of ch.:  698
 2174-2176|  2124    29 1589 | No., record pointer, word pointer of ch.:  699
 2177-2179|  2301    29 1735 | No., record pointer, word pointer of ch.:  700
 2180-2182|  2302    29 1830 | No., record pointer, word pointer of ch.:  701
 2183-2185|  2304    29 1929 | No., record pointer, word pointer of ch.:  702
 2186-2188|  2305    29 2048 | No., record pointer, word pointer of ch.:  703
 2189-2191|  2307    30   82 | No., record pointer, word pointer of ch.:  704
 2192-2194|  2312    30  136 | No., record pointer, word pointer of ch.:  705
 2195-2197|  2316    30  198 | No., record pointer, word pointer of ch.:  706
 2198-2200|  2318    30  272 | No., record pointer, word pointer of ch.:  707
 2201-2203|  2319    30  356 | No., record pointer, word pointer of ch.:  708
 2204-2206|  2321    30  412 | No., record pointer, word pointer of ch.:  709
 2207-2209|  2323    30  518 | No., record pointer, word pointer of ch.:  710
 2210-2212|  2325    30  595 | No., record pointer, word pointer of ch.:  711
 2213-2215| 20392    30  678 | No., record pointer, word pointer of ch.:  712
 ---------|------------------|-------------------------------------------------
```

```
U003003T.SN                  20-JAN-1992  11:26            PAGE  15

 ---------|------------------|-------------------------------------------------
  WORDNO. |      CONTENT     |                     COMMENT
 ---------|------------------|-------------------------------------------------
     1    |               0  | Records      length of character
     2    |              98  | Words
 ---------|------------------|-------------------------------------------------
       NAME SECTION
       ------------
     3    |               3  | Length of name section
     4    |             301  | Character number
     5    |               0  | Number of following part
 ---------|------------------|-------------------------------------------------
       SETTING INFORMATION
       -------------------
     6    |              12  | Length of setting information
     7    |               1  | Character type = letter
     8    |             198  | Number of transitions
     9    |              55  | Total set width (T=L+W+R)
    10    |               5  | Left side bearing (L)
    11    |              48  | Width (W)
    12    |               2  | Right side bearing (R)
    13    |               1  | X-minimum
    14    |              48  | X-maximum
    15    |               0  | Y-minimum
    16    |              48  | Y-maximum
    17    |               1  | Unit is 1 pixel
 ---------|------------------|-------------------------------------------------
       LINE INDEX
       ----------
    18    |               0  | Records
    19    |               2  | Words      No line index in this format
          |                  |
 ---------|------------------|-------------------------------------------------
       IMAGE INFORMATION
       -----------------
    20    |                         0  | Records   length of image information
    21    |                        79  | Words
          |                            |
          |                            | Run lengths:
   22-  23| 91 90 0                    | 10 18                            1.L.
   23-  24| 7  95 0                    | 8 21                             2.L.
   24-  25| 5  A1 0                    | 6 23                             3.L.
   25-  26| 4  A3 94 6  0              | 5 24 36 42                       4.L.
   27-  28| 3  A4 93 91 0              | 4 24 35 44                       5.L.
   29-  30| 2  A6 91 93 0              | 3 25 34 45                       6.L.
   31-  32| 2  A7 7  95 0              | 3 26 33 46                       7.L.
   32-  34| 1  B0 7  96 0              | 2 26 33 47                       8.L.
   34-  36| 1  B1 6  96 0              | 2 27 33 47                       9.L.
   36-  38| 0  94 5  91 6  97 0        | 1 13 18 27 33 48                10.L.
   38-  41| 0  92 92 6  7  96 0        | 1 11 21 27 34 48                11.L.
   41-  44| 0  90 95 6  6  91 1  4  0  | 1 9 22 28 34 43 44 48           12.L.
   44-  46| 0  7  97 5  7  7  3  4  0  | 1 8 23 28 35 42 45 49           13.L.
   46-  49| 0  7  A0 4  90 5  5  3  0  | 1 8 24 28 36 41 46 49           14.L.
   49-  51| 0  6  A1 4  A2 3  0        | 1 7 24 28 46 49                 15.L.
   51-  53| 0  6  A2 3  A2 3  0        | 1 7 25 28 46 49                 16.L.
   54-  56| 0  6  A2 4  A1 3  0        | 1 7 25 29 46 49                 17.L.
   56-  58| 0  5  A3 4  A2 3  0        | 1 6 25 29 47 50                 18.L.
   58-  60| 1  4  A4 3  A2 3  0        | 2 6 26 29 47 50                 19.L.
   60-  63| 1  5  A3 3  A2 3           | 2 7 26 29 47 50                 20.L.
 ---------|----------------------------|---------------------------------------
```

```
U003003T.SN                  20-JAN-1992  11:26           PAGE  16

---------|----------------------------|--------------------------------------
WORDNO.  |          CONTENT           |             COMMENT
---------|----------------------------|--------------------------------------
         | 1  0                       |    1 Repeated line
  63-  65| 2  4  A3 3  A1 4  0        | 3 7 26 29 46 50                 22.L.
  65-  67| 2  4  A3 4  A0 4  0        | 3 7 26 30 46 50                 23.L.
  67-  69| 3  4  A3 3  A0 4  0        | 4 8 27 30 46 50                 24.L.
  70-  72| 4  3  A3 3  A0 4  0        | 5 8 27 30 46 50                 25.L.
  72-  74| 4  4  A2 3  97 4  0        | 5 9 27 30 45 49                 26.L.
  74-  76| 5  4  A1 3  97 4  0        | 6 10 27 30 45 49                27.L.
  76-  78| 6  4  A0 4  95 5  0        | 7 11 27 31 44 49                28.L.
  79-  81| 7  4  A0 3  93 7  0        | 8 12 28 31 42 49                29.L.
  81-  83| 90 6  95 3  90 92 0        | 9 15 28 31 39 49                30.L.
  84-  84| 6  D1 0                    | 7 48                            31.L.
  85-  85| 4  D3 0                    | 5 48                            32.L.
  86-  86| 3  D3 0                    | 4 47                            33.L.
  87-  87| 2  D4 0                    | 3 47                            34.L.
  88-  88| 1  D4 0                    | 2 46                            35.L.
  89-  89| 1  D3 0                    | 2 45                            36.L.
  90-  90| 0  D2 0                    | 1 43                            37.L.
  91-  91| 0  D1 0                    | 1 42                            38.L.
  92-  92| 0  C5 0                    | 1 38                            39.L.
  93-  93| 1  7  0                    | 2 9                             40.L.
  93-  94| 1  5                       | 2 7                             41.L.
         | 1  0                       |    1 Repeated line
  94-  95| 2  4                       | 3 7                             43.L.
         | 1  0                       |    1 Repeated line
  95-  96| 3  3  0                    | 4 7                             45.L.
  96-  97| 3  4  0                    | 4 8                             46.L.
  97-  97| 4  3  0                    | 5 8                             47.L.
  98-  98| 5  3  0                    | 6 9                             48.L.
         |                            |
---------|----------------------------|--------------------------------------
   1     |                 0 | Records      length of character
   2     |                87 | Words
---------|-------------------|-------------------------------------------------
      NAME SECTION
      ------------
   3     |                 3 | Length of name section
   4     |               302 | Character number
   5     |                 0 | Number of following part
---------|-------------------|-------------------------------------------------
      SETTING INFORMATION
      -------------------
   6     |                12 | Length of setting information
   7     |                 1 | Character type = letter
   8     |               178 | Number of transitions
   9     |                63 | Total set width (T=L+W+R)
  10     |                 0 | Left side bearing (L)
  11     |                59 | Width (W)
  12     |                 4 | Right side bearing (R)
  13     |                 1 | X-minimum
  14     |                59 | X-maximum
  15     |                 0 | Y-minimum
  16     |                69 | Y-maximum
  17     |                 1 | Unit is 1 pixel
---------|-------------------|-------------------------------------------------
      LINE INDEX
      ----------
  18     |                 0 | Records
  19     |                 2 | Words      No line index in this format
         |                   |
      IMAGE INFORMATION
      -----------------
   20    |                          0 | Records   length of image information
   21    |                         68 | Words
         |                            |
         |                            | Run lengths:
  22-  23| 98 3                       | 66 69                           1.L.
         | 3  0                       |    3 Repeated lines
  23-  24| 98 4                       | 65 69                           5.L.
         | 2  0                       |    2 Repeated lines
  25-  26| 98 5  0                    | 65 70                           8.L.
  26-  27| F7 6  0                    | 64 70                           9.L.
  27-  28| F5 90 0                    | 62 70                          10.L.
  28-  29| 0  98                      | 1 70                           11.L.
         | 2  0                       |    2 Repeated lines
  30-  31| 1  98 0                    | 2 70                           14.L.
  31-  32| 2  98 0                    | 3 70                           15.L.
  32-  33| 3  98 0                    | 4 71                           16.L.
  33-  34| 4  98 0                    | 5 71                           17.L.
  35-  36| 5  98                      | 6 71                           18.L.
         | 1  0                       |    1 Repeated line
  36-  38| 4  93 A2 90 0              | 5 16 34 42                     20.L.
  38-  40| 3  91 B0 6  0              | 4 13 37 43                     21.L.
  40-  41| 3  7  B4 5  0              | 4 11 39 44                     22.L.
  41-  43| 2  6  C0 4  0              | 3 9 41 45                      23.L.
  43-  44| 2  5  C2 4  0              | 3 8 42 46                      24.L.
  44-  46| 2  4  C4 4  0              | 3 7 43 47                      25.L.
  46-  47| 1  5  C4 4  0              | 2 7 43 47                      26.L.
  47-  49| 1  4  C6 4                 | 2 6 44 48                      27.L.
         | 1  0                       |    1 Repeated line
  49-  51| 0  4  D0 4                 | 1 5 45 49                      29.L.
         | 2  0                       |    2 Repeated lines
  51-  52| 0  4  D1 3  0              | 1 5 46 49                      32.L.
  52-  54| 0  3  D2 4                 | 1 4 46 50                      33.L.
         | 1  0                       |    1 Repeated line
  54-  56| 0  4  D1 4                 | 1 5 46 50                      35.L.
         | 2  0                       |    2 Repeated lines
  56-  57| 0  4  D0 5  0              | 1 5 45 50                      38.L.
  57-  59| 0  5  C7 5  0              | 1 6 45 50                      39.L.
  59-  60| 0  5  C7 4  0              | 1 6 45 49                      40.L.
  60-  62| 1  4  C6 5  0              | 2 6 44 49                      41.L.
  62-  63| 1  5  C5 5  0              | 2 7 44 49                      42.L.
  63-  65| 1  6  C3 6  0              | 2 8 43 49                      43.L.
  65-  66| 1  7  C1 6  0              | 2 9 42 48                      44.L.
  66-  68| 2  7  B7 7  0              | 3 10 41 48                     45.L.
  68-  70| 2  91 B3 91 0              | 3 12 39 48                     46.L.
  70-  72| 2  92 B0 92 0              | 3 13 37 47                     47.L.
  72-  74| 3  94 A3 94 0              | 4 16 35 47                     48.L.
  74-  75| 4  A3 3  A3 0              | 5 24 27 46                     49.L.
  76-  76| 4  D1 0                    | 5 46                           50.L.
  77-  77| 5  C7 0                    | 6 45                           51.L.
  78-  78| 6  C5 0                    | 7 44                           52.L.
  79-  79| 7  C3 0                    | 8 43                           53.L.
  80-  81| 90 C1 0                    | 9 42                           54.L.
  81-  82| 92 B5 0                    | 11 40                          55.L.
  82-  83| 93 B3 0                    | 12 39                          56.L.
  83-  84| 95 A7 0                    | 14 37                          57.L.
  85-  86| A0 A1 0                    | 17 34                          58.L.
  86-  87| A7 3  0                    | 24 27                          59.L.
         |                            |
---------|----------------------------|---------------------------------------
```

```
U003003T.SN                  20-JAN-1992  11:26            PAGE  19

---------|------------------|-------------------------------------------------
 WORDNO. |       CONTENT    |                   COMMENT
---------|------------------|-------------------------------------------------
    1    |                 0| Records      length of character
    2    |                81| Words
---------|------------------|-------------------------------------------------
      NAME SECTION
      ------------
    3    |                 3| Length of name section
    4    |               303| Character number
    5    |                 0| Number of following part
---------|------------------|-------------------------------------------------
      SETTING INFORMATION
      -------------------
    6    |                12| Length of setting information
    7    |                 1| Character type = letter
    8    |               156| Number of transitions
    9    |                50| Total set width (T=L+W+R)
   10    |                 5| Left side bearing (L)
   11    |                43| Width (W)
   12    |                 2| Right side bearing (R)
   13    |                 1| X-minimum
   14    |                43| X-maximum
   15    |                 0| Y-minimum
   16    |                48| Y-maximum
   17    |                 1| Unit is 1 pixel
---------|------------------|-------------------------------------------------
      LINE INDEX
      ----------
   18    |                 0| Records
   19    |                 2| Words      No line index in this format
         |                  |
---------|------------------|-------------------------------------------------
      IMAGE INFORMATION
      -----------------
   20    |                            0| Records   length of image information
   21    |                           62| Words
         |                             |
         |                             | Run lengths:
  22-  23| A3 93 0                     | 20 31                           1.L.
  23-  24| 97 A3 0                     | 16 35                           2.L.
  24-  25| 94 B1 0                     | 13 38                           3.L.
  25-  26| 92 B5 0                     | 11 40                           4.L.
  27-  28| 91 B7 0                     | 10 41                           5.L.
  28-  29| 7  C2 0                     | 8 42                            6.L.
  29-  30| 6  C4 0                     | 7 43                            7.L.
  30-  31| 5  C6 0                     | 6 44                            8.L.
  31-  32| 5  C7 0                     | 6 45                            9.L.
  32-  33| 4  D1 0                     | 5 46                           10.L.
  33-  35| 3  A0 94 96 0               | 4 20 32 46                     11.L.
  35-  37| 3  94 A4 93 0               | 4 16 36 47                     12.L.
  37-  39| 2  92 B2 90 0               | 3 13 39 47                     13.L.
  39-  40| 2  91 B5 7  0               | 3 12 41 48                     14.L.
  41-  42| 1  90 C0 6  0               | 2 10 42 48                     15.L.
  42-  44| 1  7  C2 6  0               | 2 9 43 49                      16.L.
  44-  45| 1  7  C3 5  0               | 2 9 44 49                      17.L.
  45-  47| 1  6  C5 4  0               | 2 8 45 49                      18.L.
  47-  48| 0  6  C6 4  0               | 1 7 45 49                      19.L.
  48-  50| 0  6  C7 3  0               | 1 7 46 49                      20.L.
---------|-----------------------------|--------------------------------------
```



```
U003003T.SN                 20-JAN-1992  11:26            PAGE  20

 ---------|----------------------------|---------------------------------------
  WORDNO. |          CONTENT           |                COMMENT
 ---------|----------------------------|---------------------------------------
   50-  51| 0  5  D0 3  0              | 1 6 46 49                        21.L.
   51-  53| 0  5  D0 4                 | 1 6 46 50                        22.L.
          | 1  0                       |    1 Repeated line
   53-  55| 0  5  D1 3                 | 1 6 47 50                        24.L.
          | 3  0                       |    3 Repeated lines
   55-  56| 0  5  D0 4  0              | 1 6 46 50                        28.L.
   56-  58| 0  5  D0 3  0              | 1 6 46 49                        29.L.
   58-  59| 1  4  D0 3                 | 2 6 46 49                        30.L.
          | 1  0                       |    1 Repeated line
   60-  61| 1  4  B6 2  7  4  0        | 2 6 36 38 45 49                  32.L.
   62-  63| 2  4  B3 7  4  4  0        | 3 7 34 41 45 49                  33.L.
   64-  66| 2  4  B2 91 2  4  0        | 3 7 33 42 44 48                  34.L.
   66-  67| 2  5  B1 97 0              | 3 8 33 48                        35.L.
   68-  69| 3  4  B0 A0 0              | 4 8 32 48                        36.L.
   69-  71| 4  4  A7 97 0              | 5 9 32 47                        37.L.
   71-  73| 4  5  A6 97 0              | 5 10 32 47                       38.L.
   73-  74| 5  4  A7 95 0              | 6 10 33 46                       39.L.
   75-  76| 6  4  A6 94 0              | 7 11 33 45                       40.L.
   76-  78| 7  4  A5 93 0              | 8 12 33 44                       41.L.
   78-  80| 90 2  A7 91 0              | 9 11 34 43                       42.L.
   80-  81| C3 5  0                    | 36 41                            43.L.
          |                            |
 ---------|----------------------------|---------------------------------------


                             GRAND TOTAL
                             -----------
                                                    WORDS         KBYTES
                                                    -----         ------
    Length of font:                                 60090        120.180
    Length of font header:                           2215          4.430
    Length of all character headers:                13528         27.056
    Length of total image information:              44347         88.694


              DIGITIZATIONS                            NUMBER
              -------------                            ------
              Lines:                                    33795
              Total of transitions:                    128955
              -----------------------------------------------
              Characters:                                 712
              Records (a 2048 words):                      30
              -----------------------------------------------
              Raster (grid):                        100 X 100
              -----------------------------------------------
              Digitizations per line:                    3.82
              Lines per character:                         47
              Image information per character:        125 bytes


--------------------------------------------------------------------------------
```

Description of the BI format

1. Structure of a BI font

Bitmap formats are very useful for screen representation.

The structure of the BI font is as described in Apendix G with the extended structure as in Appendix L.

Word	Meaning	Comment
	Extended setting information in character record	
N1+N2+15	Number of zero bytes at the start	
N1+N2+16	Number of zero bytes at the end	New information particular to BI and GS format
N1+N2+17	Number of non zero bytes in the lines	
N1+N2+18	Number of empty lines at the start	
N1+N2+19	Number of empty lines at the end	

2. Description of the BI format

The scan identification (word number I + 18 in the font header) determines the direction a line progresses and where the following lines are.

As in the SC format the **bitmap format** is based on a rastered em square.

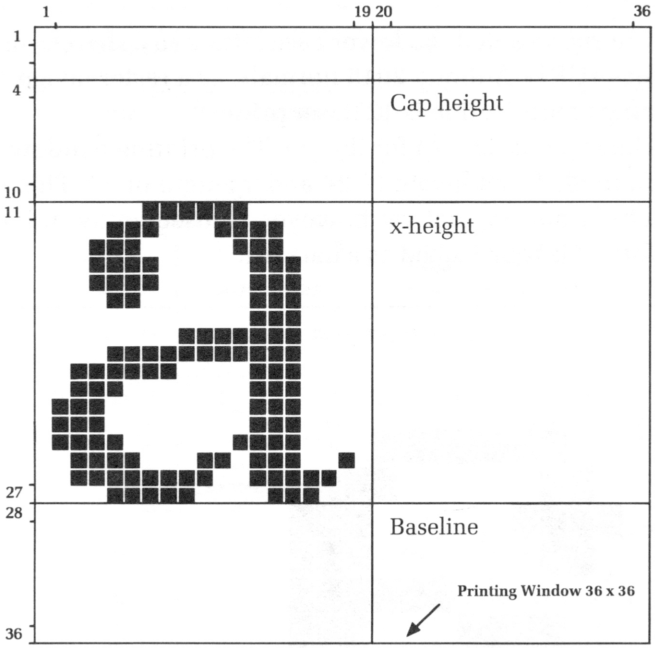
Bit value = 0 describes a white pixel, and bit value = 1 describes a black pixel.

Every sequence of 8 bits in one line is coded as a byte. In doing so, higher value bits are points further to the left in the line, lower value bits are points lying further to the right of the line. The terms left and right are geometrically correct in the case of scanning identifications 11 and 12, and in the case of identifications –11 and –12 they are correspondingly reversed. In the case of identifications 21 and 22, left means below and right means above, in identifications –21 and –22 this is correspondingly reversed.

The width and height of the printing window (word numbers I + 19 and I + 20 in the font header) give the length of the lines in bytes and the number of lines.

The number of bytes for one line is constant for all lines of any particular character. If the first bytes of all lines are equal to 0 then they are not coded as bits but rather, the number of bytes is written in the extended setting information in the character record, and similarly the number of the last bytes, which are equal to 0 in all lines. Also, the extended setting information contains the constant number of bytes for a character in which the actual image information is contained in the form of black pixels. The total number of bytes in one line is then calculated by rounding up. If, for example, in a printing window 17 dots form one line, the total number of bytes is 3. However, only the first 17 bits are relevant (see sketch).

One byte is counted as one digitization.


1
19
20
36
1
4
10
11
27
28
36
Cap height
x-height
Baseline
Printing Window 36 x 36

Example for additional setting information

Scanning code –21 results in:	
Number of zero bytes at start:	1
Number of zero bytes at end:	1
Number of non zero bytes:	3
Number of empty lines at start:	1
Number of empty lines at end:	18

Scanning code 12 results in:	
Number of zero bytes at start:	0
Number of zero bytes at end:	2
Number of non zero bytes:	3
Number of empty lines at start:	10
Number of empty lines at end:	9

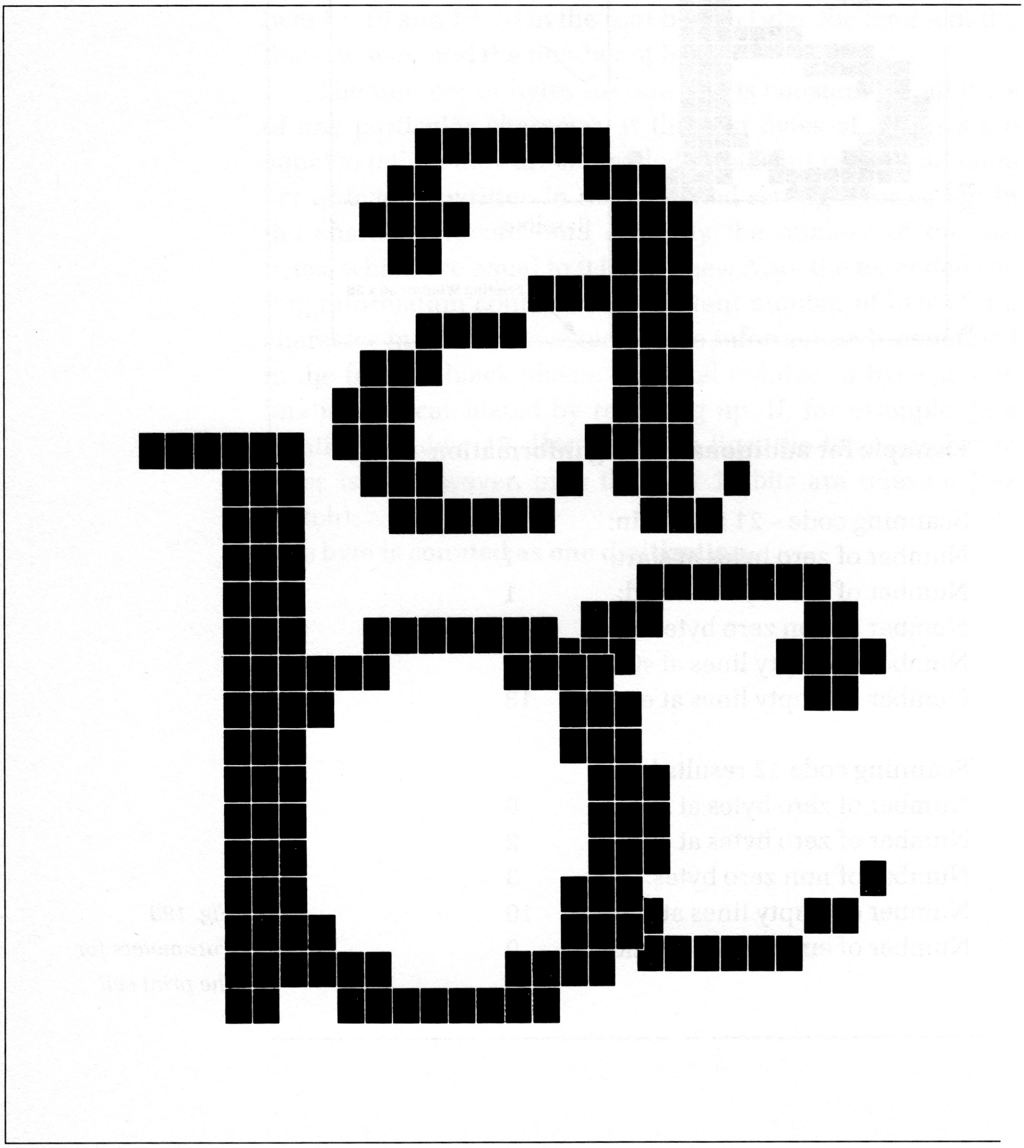
Fig. 199
Parameters for the print cell

3. Example of a BI format

First are represented the lower case letters «a», «b», «c» of the typeface «URW Antiqua 2015 normal», as a raster image. The scanning identification is 12 (laser printer).

The raster is 32 × 24 for the em. The printing window correspondingly has a height of 24 and a length of 32. This relationship is not imperative, however. Subsequently, the type font format is reproduced as a data list.

Fig. 200
BI format representation

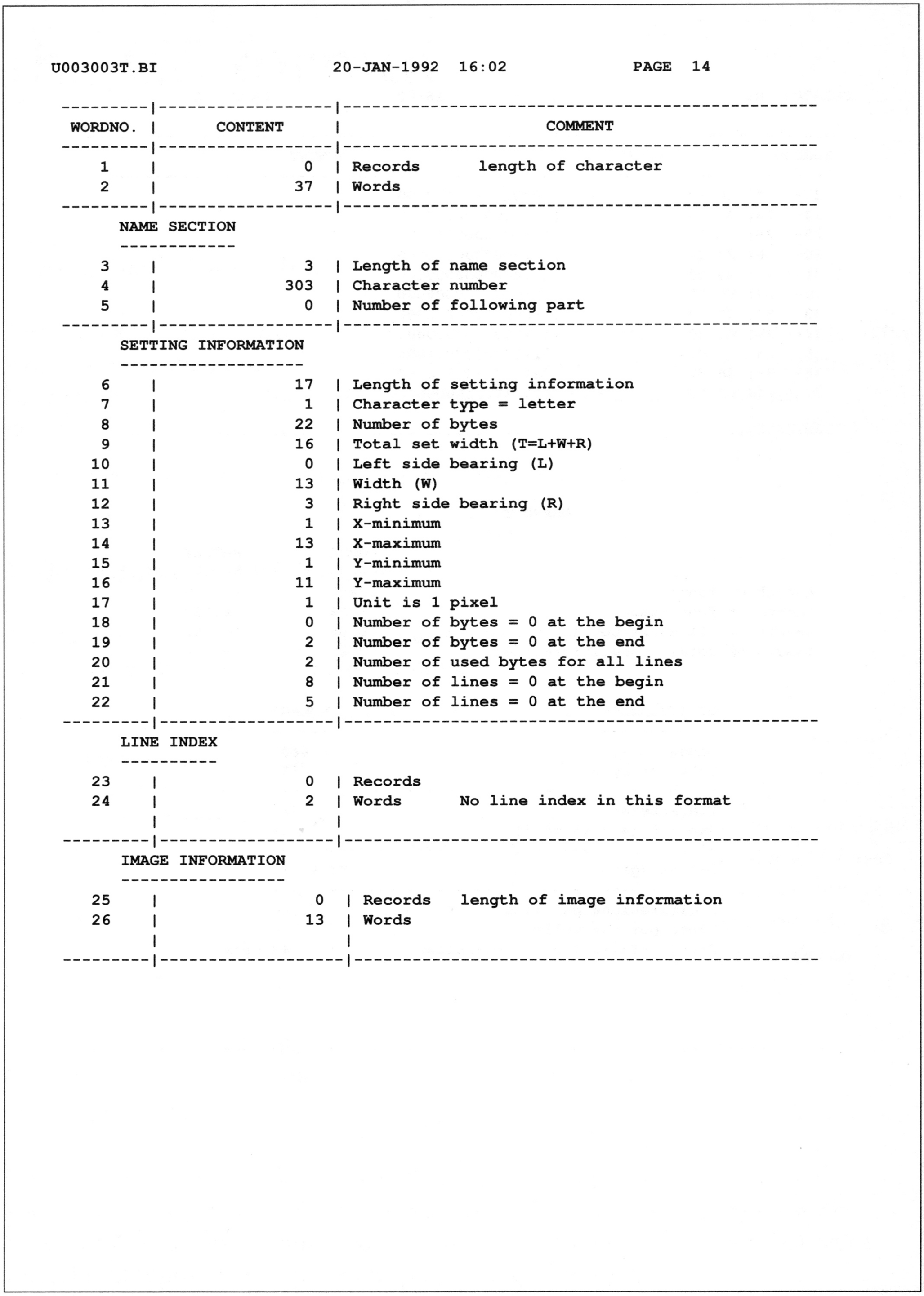


```
U003003T.BI                  20-JAN-1992  16:02             PAGE   1

---------|------------------|------------------------------------------------
 WORDNO. |      CONTENT     |                     COMMENT
---------|------------------|------------------------------------------------
    1    |              1366 | Length of font header
---------|------------------|------------------------------------------------
      NAME SECTION
      ------------
    2    |                55 | Length of name section
    3    |                 0 | Number on URW list
  4 - 10 |  U003003T.BI      | File name (ASCII)
         |                   |
 11 - 49 |URW Antiqua 2015 Regular              | Font name (ASCII)
         |
         |                   |
   50    |                BI | Data format (ASCII)
 51 - 53 |      20  1 1992   | Production date:     20-Jan-1992
 54 - 56 |      20  1 1992   | Date of last change: 20-Jan-1992
---------|------------------|------------------------------------------------
      FONT INFORMATION
      ----------------
   57    |                19 | Length of font information
   58    |                 1 | Indicator for font
   59    |               429 | Number of characters
   60    |                16 | Cap height (in pixels)
   61    |                24 | Body size (in pixels)
   62    |                11 | x-height (in pixels)
   63    |                 5 | Distance: base line - lower body line(in pixels)
   64    |                24 | Text line distance for setting (in pixels)
   65    |                 3 | Stem thickness (in pixels)
   66    |                 0 | Angle of italisation (in 1/10 degree)
   67    |                12 | Optimum point size for setting (in p)
   68    |              8462 | Average character width (in pixels)
   69    |              4688 | X-extension of a pixel (in 1/1000mm)
   70    |              6250 | Y-extension of a pixel (in 1/1000mm)
   71    |                32 | Em-resolution in X-direction (in pixels)
   72    |                24 | Em-resolution in Y-direction (in pixels)
   73    |                12 | Scan indicator of image data
   74    |                32 | Width of printing window (in pixels)
   75    |                24 | Height of printing window (in pixels)
---------|------------------|------------------------------------------------
      HIERARCHY SECTION
      -----------------
   76    |                 1 | No hierarchy section in this format
---------|------------------|------------------------------------------------
       CHARACTER INDEX
      ----------------
   77    |              1290 | Length of character index
   78    |                11 | LENGTH OF FONT IN PHYSICAL RECORDS
   79    |               370 | Last word of last record
         |                   |
   80    |               101 | Character number:  1
   81    |                 1 | Pointer to record containing start of character
   82    |              1367 | Word pointer to start of character
         |                   |
   83    |               102 | Ch. no.: 2
   84    |                 1 | Record pointer
   85    |              1417 | Word pointer
         |                   |
  86-  88|   103      1 1467 | No., record pointer, word pointer of ch.:    3
---------|------------------|------------------------------------------------
```

```
U003003T.BI                    20-JAN-1992  16:02              PAGE   2

 ---------|------------------|--------------------------------------------------
  WORDNO. |      CONTENT     |                    COMMENT
 ---------|------------------|--------------------------------------------------
   89-  91|   104      1 1519 | No., record pointer, word pointer of ch.:    4
   92-  94|   105      1 1569 | No., record pointer, word pointer of ch.:    5
   95-  97|   106      1 1619 | No., record pointer, word pointer of ch.:    6
   98- 100|   107      1 1669 | No., record pointer, word pointer of ch.:    7
  101- 103|   108      1 1721 | No., record pointer, word pointer of ch.:    8
  104- 106|   109      1 1771 | No., record pointer, word pointer of ch.:    9
  107- 109|   110      1 1813 | No., record pointer, word pointer of ch.:   10
  110- 112|   111      1 1860 | No., record pointer, word pointer of ch.:   11
  113- 115|   112      1 1910 | No., record pointer, word pointer of ch.:   12
  116- 118|   113      1 1960 | No., record pointer, word pointer of ch.:   13
  119- 121|   114      1 2018 | No., record pointer, word pointer of ch.:   14
  122- 124|   115      2   20 | No., record pointer, word pointer of ch.:   15
  125- 127|   116      2   72 | No., record pointer, word pointer of ch.:   16
  128- 130|   117      2  122 | No., record pointer, word pointer of ch.:   17
  131- 133|   118      2  192 | No., record pointer, word pointer of ch.:   18
  134- 136|   119      2  242 | No., record pointer, word pointer of ch.:   19
  137- 139|   120      2  285 | No., record pointer, word pointer of ch.:   20
  140- 142|   121      2  335 | No., record pointer, word pointer of ch.:   21



  404- 406|   285      4 1174 | No., record pointer, word pointer of ch.:  109
  407- 409|   286      4 1233 | No., record pointer, word pointer of ch.:  110
  410- 412|   287      4 1281 | No., record pointer, word pointer of ch.:  111
  413- 415|   289      4 1340 | No., record pointer, word pointer of ch.:  112
  416- 418|   290      4 1399 | No., record pointer, word pointer of ch.:  113
  419- 421|   291      4 1455 | No., record pointer, word pointer of ch.:  114
  422- 424|   292      4 1511 | No., record pointer, word pointer of ch.:  115
  425- 427|   296      4 1567 | No., record pointer, word pointer of ch.:  116
  428- 430|   297      4 1625 | No., record pointer, word pointer of ch.:  117
  431- 433|   298      4 1681 | No., record pointer, word pointer of ch.:  118
  434- 436|   301      4 1740 | No., record pointer, word pointer of ch.:  119
  437- 439|   302      4 1777 | No., record pointer, word pointer of ch.:  120
  440- 442|   303      4 1827 | No., record pointer, word pointer of ch.:  121



 1322-1324|   681     10 1909 | No., record pointer, word pointer of ch.:  415
 1325-1327|   682     10 1943 | No., record pointer, word pointer of ch.:  416
 1328-1330|   683     10 1977 | No., record pointer, word pointer of ch.:  417
 1331-1333|   684     10 2011 | No., record pointer, word pointer of ch.:  418
 1334-1336|   685     10 2045 | No., record pointer, word pointer of ch.:  419
 1337-1339|   686     11   31 | No., record pointer, word pointer of ch.:  420
 1340-1342|   687     11   65 | No., record pointer, word pointer of ch.:  421
 1343-1345|   688     11   99 | No., record pointer, word pointer of ch.:  422
 1346-1348|   689     11  133 | No., record pointer, word pointer of ch.:  423
 1349-1351|   690     11  167 | No., record pointer, word pointer of ch.:  424
 1352-1354|   691     11  201 | No., record pointer, word pointer of ch.:  425
 1355-1357|   692     11  235 | No., record pointer, word pointer of ch.:  426
 1358-1360|   693     11  269 | No., record pointer, word pointer of ch.:  427
 1361-1363|   694     11  303 | No., record pointer, word pointer of ch.:  428
 1364-1366|   695     11  337 | No., record pointer, word pointer of ch.:  429
 ---------|------------------|--------------------------------------------------
```

```
U003003T.BI                  20-JAN-1992  16:02            PAGE  10

 ---------|------------------|------------------------------------------------
  WORDNO. |      CONTENT     |                   COMMENT
 ---------|------------------|------------------------------------------------
     1    |                 0 | Records      length of character
     2    |                37 | Words
 ---------|------------------|------------------------------------------------
      NAME SECTION
      ------------

     3    |                 3 | Length of name section
     4    |               301 | Character number
     5    |                 0 | Number of following part
 ---------|------------------|------------------------------------------------
      SETTING INFORMATION
      -------------------

     6    |                17 | Length of setting information
     7    |                 1 | Character type = letter
     8    |                22 | Number of bytes
     9    |                16 | Total set width (T=L+W+R)
    10    |                 0 | Left side bearing (L)
    11    |                15 | Width (W)
    12    |                 1 | Right side bearing (R)
    13    |                 1 | X-minimum
    14    |                15 | X-maximum
    15    |                 1 | Y-minimum
    16    |                11 | Y-maximum
    17    |                 1 | Unit is 1 pixel
    18    |                 0 | Number of bytes = 0 at the begin
    19    |                 2 | Number of bytes = 0 at the end
    20    |                 2 | Number of used bytes for all lines
    21    |                 8 | Number of lines = 0 at the begin
    22    |                 5 | Number of lines = 0 at the end
 ---------|------------------|------------------------------------------------
      LINE INDEX
      ----------

    23    |                 0 | Records
    24    |                 2 | Words      No line index in this format
          |                   |
 ---------|------------------|------------------------------------------------
      IMAGE INFORMATION
      -----------------

    25    |                 0 | Records   length of image information
    26    |                13 | Words
          |                   |
 ---------|------------------|------------------------------------------------
   27-  27| 1F C0            |0001111111000000
   28-  28| 30 70            |0011000001110000
   29-  29| 78 38            |0111100000111000
   30-  30| 30 38            |0011000000111000
   31-  31| 01 F8            |0000000111111000
   32-  32| 1E 08            |0001111000001000
   33-  33| 70 08            |0111000000001000
   34-  34| E0 08            |1110000000001000
   35-  35| E0 78            |1110000001111000
   36-  36| 70 BA            |0111000010111010
   37-  37| 3F 1C            |0011111100011100
 ---------|------------------|------------------------------------------------
```

```
     U003003T.BI                  20-JAN-1992   16:02              PAGE   12

  ---------|------------------|-------------------------------------------------
   WORDNO. |      CONTENT     |                      COMMENT
  ---------|------------------|-------------------------------------------------
      1    |                0 | Records      length of character
      2    |               50 | Words
  ---------|------------------|-------------------------------------------------
        NAME SECTION
        ------------
      3    |                3 | Length of name section
      4    |              302 | Character number
      5    |                0 | Number of following part
  ---------|------------------|-------------------------------------------------
        SETTING INFORMATION
        -------------------
      6    |               17 | Length of setting information
      7    |                1 | Character type = letter
      8    |               48 | Number of bytes
      9    |               24 | Total set width (T=L+W+R)
     10    |                0 | Left side bearing (L)
     11    |               19 | Width (W)
     12    |                5 | Right side bearing (R)
     13    |                1 | X-minimum
     14    |               19 | X-maximum
     15    |                1 | Y-minimum
     16    |               16 | Y-maximum
     17    |                1 | Unit is 1 pixel
     18    |                0 | Number of bytes = 0 at the begin
     19    |                1 | Number of bytes = 0 at the end
     20    |                3 | Number of used bytes for all lines
     21    |                3 | Number of lines = 0 at the begin
     22    |                5 | Number of lines = 0 at the end
  ---------|------------------|-------------------------------------------------
        LINE INDEX
        ----------
     23    |                0 | Records
     24    |                2 | Words      No line index in this format
           |                  |
  ---------|------------------|-------------------------------------------------
        IMAGE INFORMATION
        -----------------
     25    |                 0 | Records   length of image information
     26    |                26 | Words
           |                  |
  ---------|-------------------|------------------------------------------------
    27-  28| FC 00 00          |111111000000000000000000
    28-  29| 1C 00 00          |000111000000000000000000
    30-  31| 1C 00 00          |000111000000000000000000
    31-  32| 1C 00 00          |000111000000000000000000
    33-  34| 1C 00 00          |000111000000000000000000
    34-  35| 1C FE 00          |000111001111111000000000
    36-  37| 1F 87 80          |000111111000011110000000
    37-  38| 1E 01 C0          |000111100000000111000000
    39-  40| 1C 01 C0          |000111000000000111000000
    40-  41| 1C 00 E0          |000111000000000011100000
    42-  43| 1C 00 E0          |000111000000000011100000
    43-  44| 1C 00 E0          |000111000000000011100000
    45-  46| 1C 01 C0          |000111000000000111000000
    46-  47| 1E 01 C0          |000111100000000111000000
    48-  49| 1F 87 80          |000111111000011110000000
    49-  50| 19 FE 00          |000110011111111000000000
  ---------|-------------------|------------------------------------------------
```



```
U003003T.BI                    20-JAN-1992  16:02             PAGE  14

---------|------------------|--------------------------------------------------
 WORDNO. |      CONTENT     |                         COMMENT
---------|------------------|--------------------------------------------------
    1    |                0 | Records      length of character
    2    |               37 | Words
---------|------------------|--------------------------------------------------
      NAME SECTION
      ------------
    3    |                3 | Length of name section
    4    |              303 | Character number
    5    |                0 | Number of following part
---------|------------------|--------------------------------------------------
      SETTING INFORMATION
      -------------------
    6    |               17 | Length of setting information
    7    |                1 | Character type = letter
    8    |               22 | Number of bytes
    9    |               16 | Total set width (T=L+W+R)
   10    |                0 | Left side bearing (L)
   11    |               13 | Width (W)
   12    |                3 | Right side bearing (R)
   13    |                1 | X-minimum
   14    |               13 | X-maximum
   15    |                1 | Y-minimum
   16    |               11 | Y-maximum
   17    |                1 | Unit is 1 pixel
   18    |                0 | Number of bytes = 0 at the begin
   19    |                2 | Number of bytes = 0 at the end
   20    |                2 | Number of used bytes for all lines
   21    |                8 | Number of lines = 0 at the begin
   22    |                5 | Number of lines = 0 at the end
---------|------------------|--------------------------------------------------
      LINE INDEX
      ----------
   23    |                0 | Records
   24    |                2 | Words      No line index in this format
         |                  |
---------|------------------|--------------------------------------------------
      IMAGE INFORMATION
      -----------------
   25    |                 0 | Records   length of image information
   26    |                13 | Words
         |                   |
---------|-------------------|-------------------------------------------------
```

```
U003003T.BI                   20-JAN-1992  16:02             PAGE  15

 ---------|--------------------|-----------------------------------------------
  WORDNO. |       CONTENT      |                    COMMENT
 ---------|--------------------|-----------------------------------------------
   27-  27| 0F E0              |0000111111100000
   28-  28| 38 30              |0011100000110000
   29-  29| 70 78              |0111000001111000
   30-  30| E0 30              |1110000000110000
   31-  31| E0 00              |1110000000000000
   32-  32| E0 00              |1110000000000000
   33-  33| E0 00              |1110000000000000
   34-  34| E0 00              |1110000000000000
   35-  35| 70 08              |0111000000001000
   36-  36| 38 30              |0011100000110000
   37-  37| 0F C0              |0000111111000000
          |                    |
 ---------|--------------------|-----------------------------------------------


                              GRAND TOTAL
                              -----------
                                                  WORDS          KBYTES
                                                  -----          ------
    Length of font:                               20850          41.700
    Length of font header:                         1366           2.732
    Length of all character headers:              10296          20.592
    Length of total image information:             9188          18.376

              DIGITIZATIONS                           NUMBER
              -------------                           ------
              Lines:                                    6681
              Total of bytes:                          16537
              ----------------------------------------------
              Characters:                                429
              Records (a 2048 words):                     11
              ----------------------------------------------
              Raster (grid):                         32 X  24
              ----------------------------------------------
              Digitizations per line:                   2.48
              Lines per character:                        16
              Image information per character:            43 bytes



--------------------------------------------------------------------------------
```

Description of the GS format

1. Structure of a GS font

The structure of a grayscale (GS) font is as described in Appendix G with an extended structure as in Appendix L. In the identification for a font (word number 50) the number of gray tones is coded:

G1 for 2 bit, G2 for 3 bit, G3 for 4 bit, G4 for 8 bit etc.

Conceived for future color screens with gray scale representation of text

2. Description of the GS format

The **grayscale format** was designed to describe points which are not only black or white but also halftone.

In the case of a 4 bit coding with linear interpretation for gray the following gray scales are possible:

Hexadecimal 4 bit value	Black fraction in %	Comment
0	0	white
1	7	.
2	16	light
.	.	.
.	.	.
.	.	.
E	93	dark
F	100	black

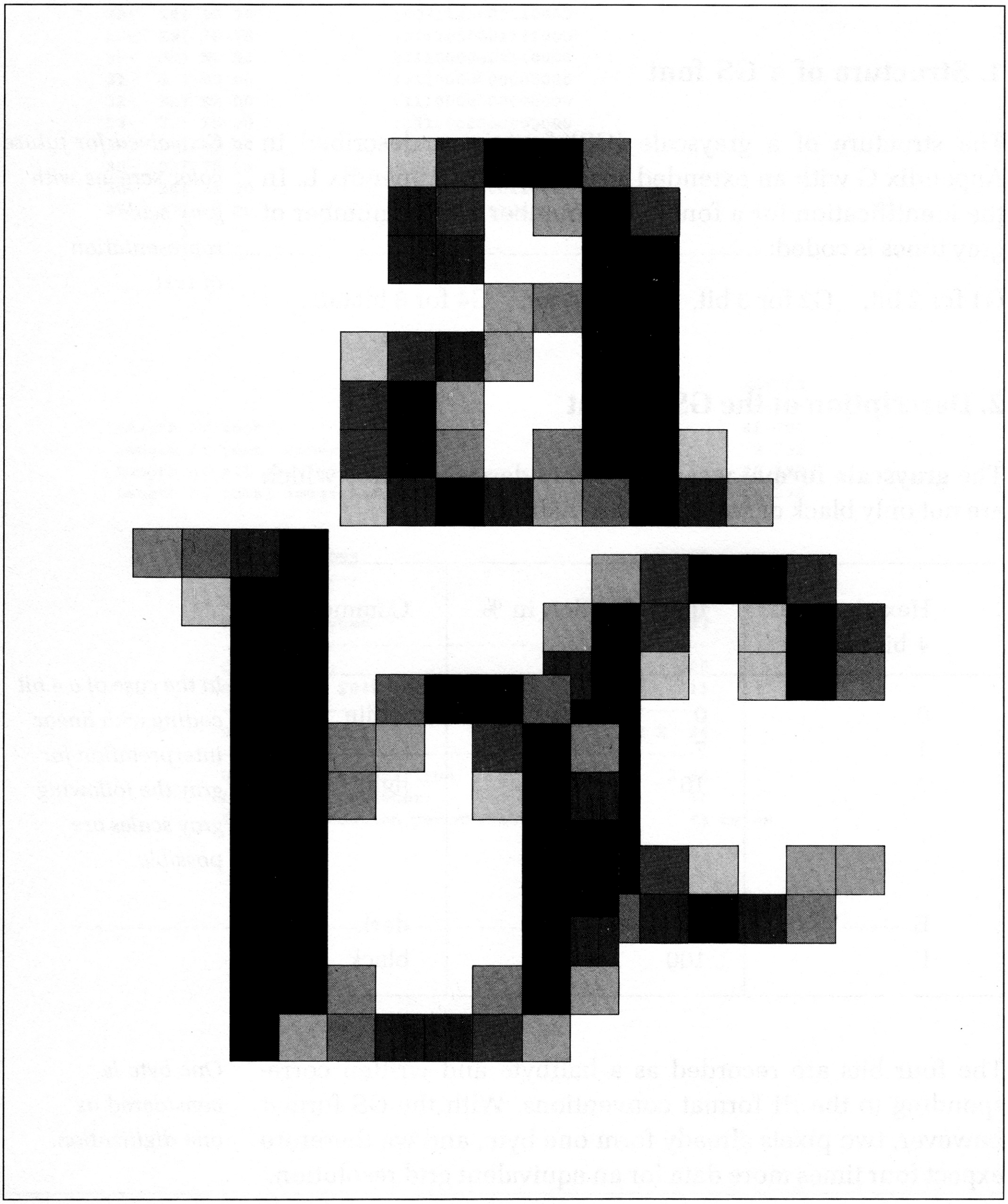
The four bits are recorded as a halfbyte and written corresponding to the BI format conventions. With the GS format however, two pixels already form one byte, and we therefore expect four times more data for an equivalent grid resolution.

One byte is considered as one digitization.

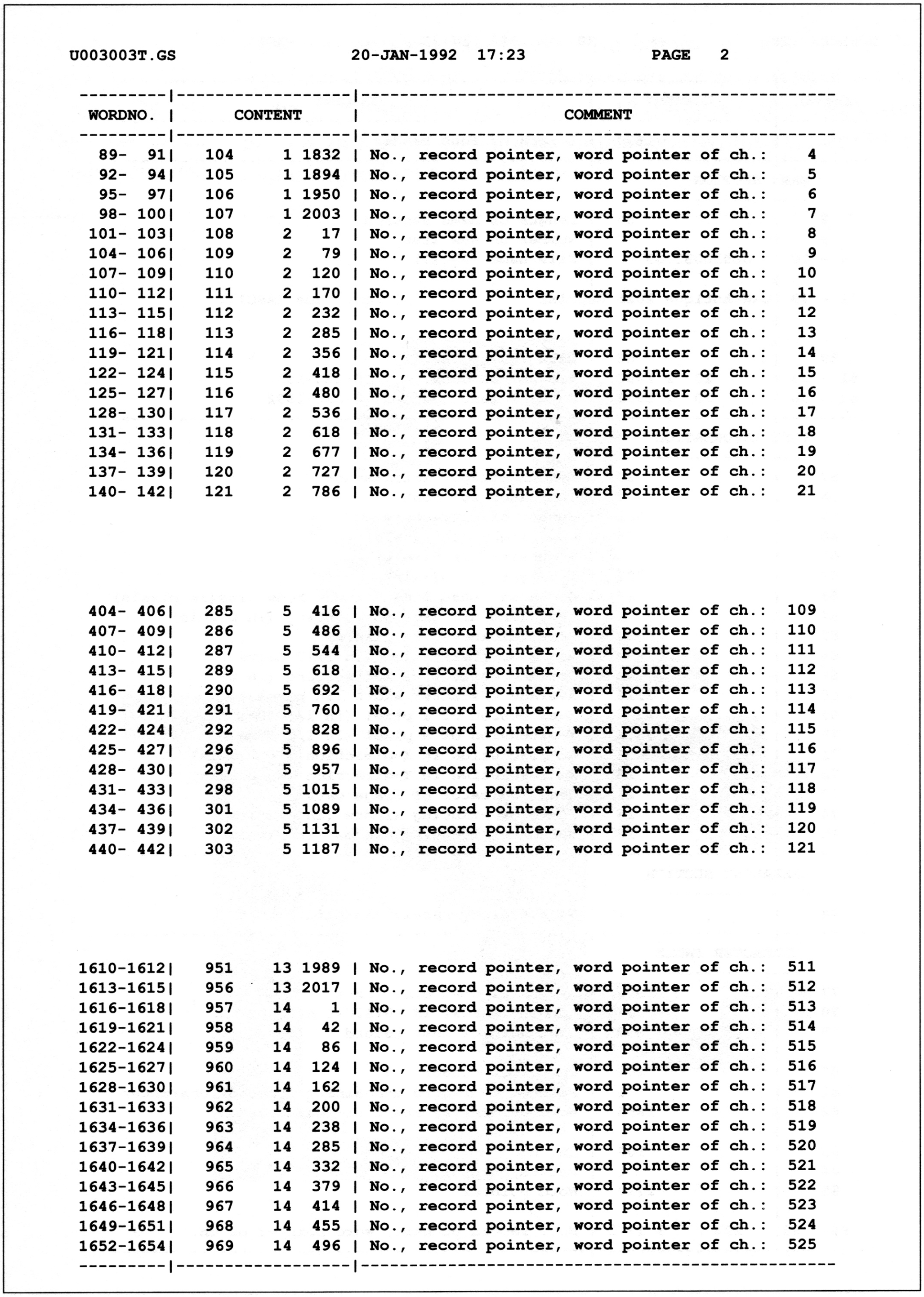
3. Example of a GS format

Fig. 201
GS format representation

First we represent the lower case letters «a», «b», «c» of the typeface «URW Antiqua 2015 normal», as graymaps of a 4 bit coding. The scan identification is 12 (laser printer). The raster is 16 × 16 for the em square. Subsequently, the type font format is reproduced as a data list.



```
U003003T.GS                  20-JAN-1992  17:23              PAGE   1

 ---------|------------------|------------------------------------------------
  WORDNO. |       CONTENT    |                    COMMENT
 ---------|------------------|------------------------------------------------
     1    |            1654  | Length of font header
 ---------|------------------|------------------------------------------------
       NAME SECTION
       ------------
     2    |              55  | Length of name section
     3    |               0  | Number on URW list
   4 - 10 |  U003003T.G3     | File name (ASCII)
          |                  |
  11 - 49 |URW Antiqua 2015 Regular              | Font name (ASCII)
          |
          |                  |
    50    |              G3  | Data format (ASCII)
  51 - 53 |      20  1 1992  | Production date:     20-Jan-1992
  54 - 56 |      20  1 1992  | Date of last change: 20-Jan-1992
 ---------|------------------|------------------------------------------------
       FONT INFORMATION
       ----------------
    57    |              19  | Length of font information
    58    |               1  | Indicator for font
    59    |             525  | Number of characters
    60    |              29  | Cap height (in pixels)
    61    |               1  | Body size (in pixels)
    62    |             327  | x-height (in pixels)
    63    |              29  | Distance: base line - lower body line(in pixels)
    64    |               1  | Text line distance for setting (in pixels)
    65    |             352  | Stem thickness (in pixels)
    66    |               0  | Angle of italisation (in 1/10 degree)
    67    |              12  | Optimum point size for setting (in p)
    68    |            2116  | Average character width (in pixels)
    69    |            9376  | X-extension of a pixel (in 1/1000mm)
    70    |            9376  | Y-extension of a pixel (in 1/1000mm)
    71    |              16  | Em-resolution in X-direction (in pixels)
    72    |              16  | Em-resolution in Y-direction (in pixels)
    73    |              21  | Scan indicator of image data
    74    |              16  | Width of printing window (in pixels)
    75    |              16  | Height of printing window (in pixels)
 ---------|------------------|------------------------------------------------
       HIERARCHY SECTION
       -----------------
    76    |               1  | No hierarchy section in this format
 ---------|------------------|------------------------------------------------
        CHARACTER INDEX
       ----------------
    77    |            1578  | Length of character index
    78    |              14  | LENGTH OF FONT IN PHYSICAL RECORDS
    79    |             539  | Last word of last record
          |                  |
    80    |             101  | Character number:  1
    81    |               1  | Pointer to record containing start of character
    82    |            1655  | Word pointer to start of character
          |                  |
    83    |             102  | Ch. no.: 2
    84    |               1  | Record pointer
    85    |            1717  | Word pointer
          |                  |
   86-  88|   103     1 1773 | No., record pointer, word pointer of ch.:    3
 ---------|------------------|------------------------------------------------
```

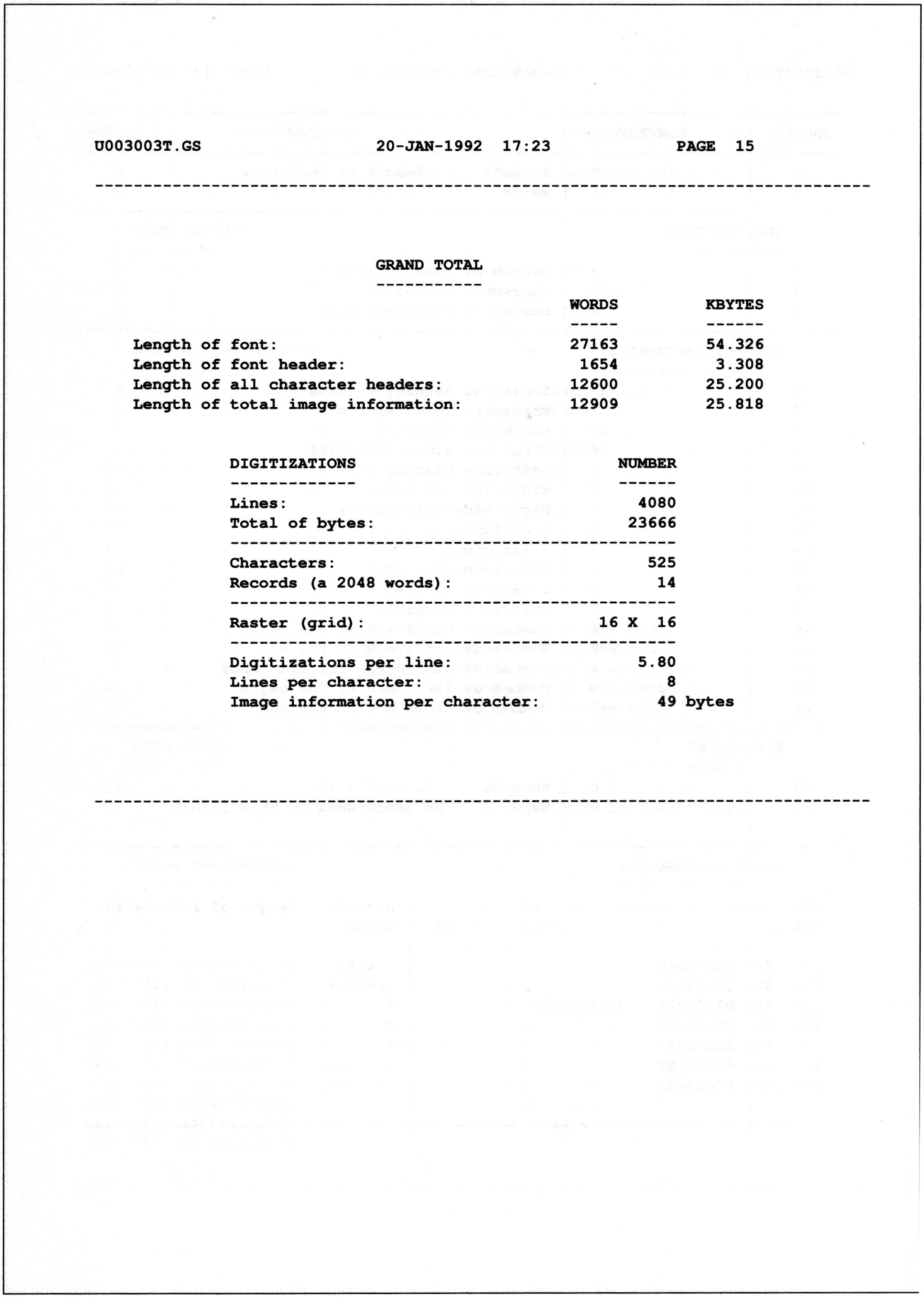


```
U003003T.GS                    20-JAN-1992  17:23             PAGE    2

 ---------|------------------|---------------------------------------------
  WORDNO. |     CONTENT      |                   COMMENT
 ---------|------------------|---------------------------------------------
   89-  91|   104      1 1832 | No., record pointer, word pointer of ch.:    4
   92-  94|   105      1 1894 | No., record pointer, word pointer of ch.:    5
   95-  97|   106      1 1950 | No., record pointer, word pointer of ch.:    6
   98- 100|   107      1 2003 | No., record pointer, word pointer of ch.:    7
  101- 103|   108      2   17 | No., record pointer, word pointer of ch.:    8
  104- 106|   109      2   79 | No., record pointer, word pointer of ch.:    9
  107- 109|   110      2  120 | No., record pointer, word pointer of ch.:   10
  110- 112|   111      2  170 | No., record pointer, word pointer of ch.:   11
  113- 115|   112      2  232 | No., record pointer, word pointer of ch.:   12
  116- 118|   113      2  285 | No., record pointer, word pointer of ch.:   13
  119- 121|   114      2  356 | No., record pointer, word pointer of ch.:   14
  122- 124|   115      2  418 | No., record pointer, word pointer of ch.:   15
  125- 127|   116      2  480 | No., record pointer, word pointer of ch.:   16
  128- 130|   117      2  536 | No., record pointer, word pointer of ch.:   17
  131- 133|   118      2  618 | No., record pointer, word pointer of ch.:   18
  134- 136|   119      2  677 | No., record pointer, word pointer of ch.:   19
  137- 139|   120      2  727 | No., record pointer, word pointer of ch.:   20
  140- 142|   121      2  786 | No., record pointer, word pointer of ch.:   21




  404- 406|   285      5  416 | No., record pointer, word pointer of ch.:  109
  407- 409|   286      5  486 | No., record pointer, word pointer of ch.:  110
  410- 412|   287      5  544 | No., record pointer, word pointer of ch.:  111
  413- 415|   289      5  618 | No., record pointer, word pointer of ch.:  112
  416- 418|   290      5  692 | No., record pointer, word pointer of ch.:  113
  419- 421|   291      5  760 | No., record pointer, word pointer of ch.:  114
  422- 424|   292      5  828 | No., record pointer, word pointer of ch.:  115
  425- 427|   296      5  896 | No., record pointer, word pointer of ch.:  116
  428- 430|   297      5  957 | No., record pointer, word pointer of ch.:  117
  431- 433|   298      5 1015 | No., record pointer, word pointer of ch.:  118
  434- 436|   301      5 1089 | No., record pointer, word pointer of ch.:  119
  437- 439|   302      5 1131 | No., record pointer, word pointer of ch.:  120
  440- 442|   303      5 1187 | No., record pointer, word pointer of ch.:  121




 1610-1612|   951     13 1989 | No., record pointer, word pointer of ch.:  511
 1613-1615|   956     13 2017 | No., record pointer, word pointer of ch.:  512
 1616-1618|   957     14    1 | No., record pointer, word pointer of ch.:  513
 1619-1621|   958     14   42 | No., record pointer, word pointer of ch.:  514
 1622-1624|   959     14   86 | No., record pointer, word pointer of ch.:  515
 1625-1627|   960     14  124 | No., record pointer, word pointer of ch.:  516
 1628-1630|   961     14  162 | No., record pointer, word pointer of ch.:  517
 1631-1633|   962     14  200 | No., record pointer, word pointer of ch.:  518
 1634-1636|   963     14  238 | No., record pointer, word pointer of ch.:  519
 1637-1639|   964     14  285 | No., record pointer, word pointer of ch.:  520
 1640-1642|   965     14  332 | No., record pointer, word pointer of ch.:  521
 1643-1645|   966     14  379 | No., record pointer, word pointer of ch.:  522
 1646-1648|   967     14  414 | No., record pointer, word pointer of ch.:  523
 1649-1651|   968     14  455 | No., record pointer, word pointer of ch.:  524
 1652-1654|   969     14  496 | No., record pointer, word pointer of ch.:  525
 ---------|------------------|---------------------------------------------
```

```
U003003T.GS                  20-JAN-1992  17:23             PAGE  12

 ---------|------------------|--------------------------------------------------
  WORDNO. |      CONTENT     |                      COMMENT
 ---------|------------------|--------------------------------------------------
     1    |               0  | Records      length of character
     2    |              42  | Words
 ---------|------------------|--------------------------------------------------
       NAME SECTION
       ------------
     3    |               3  | Length of name section
     4    |             301  | Character number
     5    |               0  | Number of following part
 ---------|------------------|--------------------------------------------------
       SETTING INFORMATION
       -------------------
     6    |              17  | Length of setting information
     7    |               1  | Character type = letter
     8    |              32  | Number of bytes
     9    |               9  | Total set width (T=L+W+R)
    10    |               1  | Left side bearing (L)
    11    |               8  | Width (W)
    12    |               1  | Right side bearing (R)
    13    |               1  | X-minimum
    14    |               8  | X-maximum
    15    |               1  | Y-minimum
    16    |               8  | Y-maximum
    17    |               1  | Unit is 1 pixel
    18    |               4  | Number of bytes = 0 at the begin
    19    |               0  | Number of bytes = 0 at the end
    20    |               4  | Number of used bytes for all lines
    21    |               0  | Number of lines = 0 at the begin
    22    |               8  | Number of lines = 0 at the end
 ---------|------------------|--------------------------------------------------
       LINE INDEX
       ----------
    23    |               0  | Records
    24    |               2  | Words      No line index in this format
          |                  |
 ---------|------------------|--------------------------------------------------
       IMAGE INFORMATION
       -----------------
    25    |                               0  | Records   length of information
    26    |                              18  | Words
          |                                  |
   27-  28| 3EF90571                         | .X#+ :-.
   29-  30| DD8D3BF8                         | XX+X.*#+
   31-  32| E1028127                         | X. .+..-
   33-  34| A1009007                         | *.  +  -
   35-  36| 1B44A46B                         | .*::*:-*
   37-  38| BFFFFFF5                         | *######:
   39-  40| B5444410                         | *:::::.
   41-  42| 32000000                         | ..
          |                                  |
 ---------|----------------------------------|----------------------------------
```

```
U003003T.GS                  20-JAN-1992  17:23             PAGE  13

 ---------|------------------|-------------------------------------------------
  WORDNO. |      CONTENT     |                    COMMENT
 ---------|------------------|-------------------------------------------------
     1    |               0  | Records      length of character
     2    |              56  | Words
 ---------|------------------|-------------------------------------------------
       NAME SECTION
       ------------
     3    |               3  | Length of name section
     4    |             302  | Character number
     5    |               0  | Number of following part
 ---------|------------------|-------------------------------------------------
       SETTING INFORMATION
       -------------------
     6    |              17  | Length of setting information
     7    |               1  | Character type = letter
     8    |              60  | Number of bytes
     9    |              11  | Total set width (T=L+W+R)
    10    |               0  | Left side bearing (L)
    11    |              10  | Width (W)
    12    |               1  | Right side bearing (R)
    13    |               1  | X-minimum
    14    |              10  | X-maximum
    15    |               1  | Y-minimum
    16    |              12  | Y-maximum
    17    |               1  | Unit is 1 pixel
    18    |               4  | Number of bytes = 0 at the begin
    19    |               0  | Number of bytes = 0 at the end
    20    |               6  | Number of used bytes for all lines
    21    |               0  | Number of lines = 0 at the begin
    22    |               6  | Number of lines = 0 at the end
 ---------|------------------|-------------------------------------------------
       LINE INDEX
       ----------
    23    |               0  | Records
    24    |               2  | Words      No line index in this format
          |                  |
 ---------|------------------|-------------------------------------------------
       IMAGE INFORMATION
       -----------------
    25    |                               0  | Records   length of information
    26    |                              32  | Words
          |                                  |
   27-  29| 000000000090                     |           +
   30-  32| 4444444447F1                     | :::::::::-#.
   33-  35| 6FFFFFFFFFF4                     | -##########:
   36-  38| 882002A30000                     | ++.  .*.
   39-  41| A00000090000                     | *      +
   42-  44| C000000A0000                     | X      *
   45-  47| C100001B0000                     | X.    .*
   48-  50| 8E7447E80000                     | +X-::-X+
   51-  53| 09FFFFA00000                     |  +####*
   54-  56| 002442000000                     |   .::.
          |                                  |
 ---------|----------------------------------|--------------------------------
```

```
U003003T.GS                  20-JAN-1992  17:23             PAGE  14

 ---------|------------------|-------------------------------------------------
  WORDNO. |      CONTENT     |                     COMMENT
 ---------|------------------|-------------------------------------------------
     1    |               0  | Records      length of character
     2    |              40  | Words
 ---------|------------------|-------------------------------------------------
       NAME SECTION
       ------------
     3    |               3  | Length of name section
     4    |             303  | Character number
     5    |               0  | Number of following part
 ---------|------------------|-------------------------------------------------
       SETTING INFORMATION
       -------------------
     6    |              17  | Length of setting information
     7    |               1  | Character type = letter
     8    |              28  | Number of bytes
     9    |               8  | Total set width (T=L+W+R)
    10    |               1  | Left side bearing (L)
    11    |               7  | Width (W)
    12    |               1  | Right side bearing (R)
    13    |               1  | X-minimum
    14    |               7  | X-maximum
    15    |               1  | Y-minimum
    16    |               8  | Y-maximum
    17    |               1  | Unit is 1 pixel
    18    |               4  | Number of bytes = 0 at the begin
    19    |               0  | Number of bytes = 0 at the end
    20    |               4  | Number of used bytes for all lines
    21    |               0  | Number of lines = 0 at the begin
    22    |               9  | Number of lines = 0 at the end
 ---------|------------------|-------------------------------------------------
       LINE INDEX
       ----------
    23    |               0  | Records
    24    |               2  | Words      No line index in this format
          |                  |
 ---------|------------------|-------------------------------------------------
       IMAGE INFORMATION
       -----------------
    25    |                               0  | Records   length of information
    26    |                              16  | Words
          |                                  |
   27-  28| 04DFFD40                         |  :X##X:
   29-  30| 4FDAADF4                         | :#X**X#:
   31-  32| B700003A                         | *-    .*
   33-  34| E0000008                         | X      +
   35-  36| A0000007                         | *      -
   37-  38| 82000CB9                         | +.   X*+
   39-  40| 170008B1                         | .-   +*.
          |                                  |
 ---------|----------------------------------|---------------------------------
```



```
U003003T.GS                  20-JAN-1992  17:23              PAGE  15
-------------------------------------------------------------------------------


                             GRAND TOTAL
                             -----------
                                                   WORDS          KBYTES
                                                   -----          ------
    Length of font:                                27163          54.326
    Length of font header:                          1654           3.308
    Length of all character headers:               12600          25.200
    Length of total image information:             12909          25.818


              DIGITIZATIONS                             NUMBER
              -------------                             ------
              Lines:                                      4080
              Total of bytes:                            23666
              ------------------------------------------------
              Characters:                                  525
              Records (a 2048 words):                       14
              ------------------------------------------------
              Raster (grid):                           16 X  16
              ------------------------------------------------
              Digitizations per line:                     5.80
              Lines per character:                           8
              Image information per character:              49 bytes



-------------------------------------------------------------------------------
```

Data structure for spacing

1. General

URW currently has four different specifications of letter spacing available:

- traditional (width tables),
- kerning,
- touching, and
- overlapping.

Traditional widths are stored in the form of width tables. For kerning there is a short and a long form of storage. The short form is created from classifications of the left and right side appearance of all characters of any font in question. URW uses the clustering method which classifies similar side appearances into classes and which allows, for example, 32 different classes of right side appearance (back classes). The same applies to the left side appearances (front classes). This produces a short kerning table of 32 × 32 values for all possible combinations of left and right side classes.

In the long kerning table all character pairs have individual correction values, i.e. the long table has 100 × 100 values when applied to 100 different characters, and 250 × 250 values when applied to a typeface of 250 characters.

To date we have utilized the long storage form also for touching and overlapping tables.

2. Access

Access to letter spacing is organized exactly as for a font, i.e. we acquire the data structure of a font (Appendix G) and adjust the contents for the purpose of letter spacing. The widths are known as «character number 1», the short table for kerning as «character number 2», the long table as «number 3» and touching and overlapping tables as «number 4» and «5».

For additional letter spacing variations there is ample accomodation from number 6 onwards.

One correction value is stored as 16 bit integer and is regarded as *one digitization.*

The correction values are accessed through a changed contour index as follows:

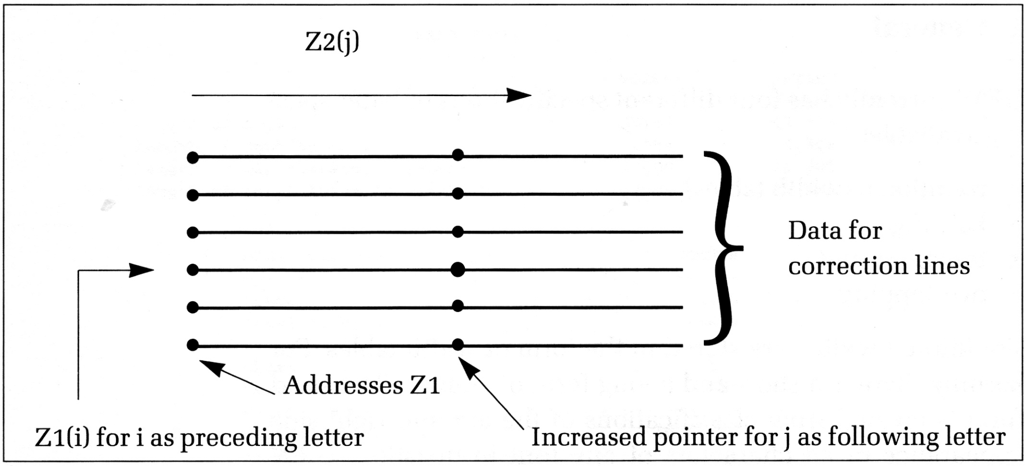

Z2(j)
Data for correction lines
Addresses Z1
Z1(i) for i as preceding letter
Increased pointer for j as following letter

Fig. 202
The correction values are accessed through a changed contour index.

For character number i in position i the index contains the start address Z1(i) of the correction line (old image information) with the correction values for letter number i as preceding letter. This first section of the index has a length of 2 × N4 (in words). In addition, the letter spacing index contains for letter number j in position 2 × N4 + j the address Z2(j) in the correction lines, where the correction value for the letter number j is stored as the following letter. This is the second section of the index. For character pair (ij) the correction value for the letter spacing is found in position Z1(i) + Z2(j) in the image information.

The length of the index is dependent upon the highest character number of the respective font. Our current concept allows space for a maximum length of 1000 places per index table. However, this length is a variable and can be adjusted to meet individual needs. There is a default correction value of 0. This correction has no effect and is assigned to all letter pairs which are not intended to kern, touch, or overlap.

3. Description of the letter spacing header

In this appendix is described letter spacing using a changed font header. The data records for the tables follow in Appendices T, U and V.

Data carriers are as described in Appendix G, point 1.

Spacing header

Word	Meaning	Comment
1	Length of spacing header in words	= 1 + I + J + L + M
	Name section	
2	Length of name section	(I)
3	Number on URW list	
4 – 9	File name	6 words
10 – 49	Font name in ASCII	40 words
50	Format identification of data	SP (spacing)
51 – 53	Production date	3 words
54 – 56	Date of last change	e.g. 16,6,92 for 16th of June 1992. Currently (1992) I has the value 55 at URW.
	Spacing information	
I + 2	Length of spacing data in words	(J)
I + 3	Identification of spacing	= 3
I + 4	Number of tables	Currently = 5
I + 5	Cap heights	in units At URW currently (1992) J has the value 4.

Word	Meaning	Comment
	Hierarchy section	
I + J + 2	Length of hierarchy	(L) =1 for spacing
	Table index	
I + J + L +2	Length of table index	(M)
I + J + L + 3	Pointer to last written physical record (=**length of spacing in physical records**)	Current length of spacing data
I + J + L + 4	Pointer to last written word of last record	
I + J + L + 5	Number of first table	Widths
I + J + L + 6	Record pointer in the spacing which contains the start of the first table	
I + J + L + 7	Pointer to the first word used in this record	
I + J + L + 8	Number	Short kerning (clustering)
I + J + L + 9	Record pointer	
I + J + L + 10	Word pointer	
I + J + L + 11	Number	Long kerning
I + J + L + 12	Record pointer	
I + J + L + 13	Word pointer	
I + J + L + 14	Number	Touching
I + J + L + 15	Record pointer	
I + J + L + 16	Word pointer	
I + J + L + 17	Number	Overlapping
I + J + L + 18	Record pointer	
I + J + L + 19	Word pointer	

Adjustment of letter spacing for all data records can only be accomplished together with the associated character number list (see Appendix B) of the respective font.

Width table

Width tables are compressible as one needs no access to a matrix, but rather to a simple array. This exception will be ignored here. For this reason the width table in letter spacing (see Appendix S) has the structure shown below.

Width record

Word	Meaning	Comment
	Length of width record	2 + N1 + N2 + N3 + M
1	in records	
2	in words	
	Name section	
3	Length of name section	N1 (currently = 3)
4	Number = 1 (width)	
5	Following part number	= 0 (currently)
	Record information	
N1 + 3	Length of record information	N2 (currently = 3)
N1 + 4	Number of widths	
N1 + 5	Unit size	according to font format
	Spacing index	
	Length of letter spacing index	N3 = 2 + N4 × 3
N1 + N2 + 3	in records	
N1 + N2 + 4	in words	
N1 + N2 + 5	Record and word pointer of	The adresses are counted relative to the start of spacing information.
N1 + N2 + 6	width of character number 1	

Word	Meaning	Comment
N1 + N2 + 7 N1 + N2 + 8	Record and word pointer of width of character number 2	N4 entries for width line ≙ Adresses Z2(j) as per appendix R,
. . .	. .	
N1 + N2 + 4 + N4 × 2	Word pointer of width of highest character number of font	
N1 + N2 + 5 + N4 × 2 . .	0 .	N4 entries for width lines ≙ Addresses Z2(j) as per appendix R, all N4 entries are given the value 0.
	Letter spacing information	
N1 + N2 + N3	Length of table	M
+ 3	in records	
+ 4	in words	
+ 5	«Width» of the em	Default parameter
+ 6	Width of first character with no default width	
+ 7	Width of second character with no default width	
. . .	. . .	
N1 + N2 + N3 + M + 2	Last width	

An example of a width table record is shown at the end of appendix T, the table is representing traditional widths for the typeface «URW Antiqua 2015 normal».

To read the width of character i and the next character j from the table we have the following:

Address: $= Z1(i) + Z2(j)$, Width: $=$ contents (address).

Details of the next character «j» are then irrelevant. We do however intend to adjust access to widths, to that of the method for kerning correction values.

Appendix U

Kerning table

Here are long and short versions of kerning tables. As mentioned earlier, the short tables are generated by classifying left and right side appearances of characters, and long tables as kerning corrections for all possible letter pairs in a type font. The short kerning table has the following structure (see also Appendix S):

There is short kerning and long kerning.

1. Short kerning record

Word	Meaning	Comment
	Length of short kerning record	2+N1+N2+N3+M
1	in records	
2	in words	
	Name section	
3	Length of name section	N1 (currently = 3)
4	Number = 2	(short kerning)
5	Following part number	
	Record information	
N1 + 3	Length of record information	N2 (currently = 4)
N1 + 4	Number of correction values	
N1 + 5	Unit size	According to font format
N1 + 6	Number of corresponding long kerning table	
	Spacing index	
	Length of spacing index	N3 = 2 + N4 × 3
N1 + N2 + 3	in records	

Word	Meaning	Comment
N1 + N2 + 4	in words	
N1 + N2 + 5 N1 + N2 + 6	Record and word pointer to correction line for letter pairs, in which the first (front) letter is character number 1.	The addresses are counted relative to start of spacing information.
N1 + N2 + 7 N1 + N2 + 8	Record and word pointer to correction line of the letter pair in which the first (front) letter is character number 2.	N4 entries for correction lines ≙ Addresses Z1(i) as per appendix R
. . .	. . .	
N1 + N2 + 4 + N4 × 2	Word pointer to correction line of the letter pair in which the last character is in front.	
N1 + N2 + 5 + N4 × 2	Word pointer to the place in the correction line for letter pairs in which character number 1 is back.	N4 entries for correction lines ≙ Addresses Z2(j) as per appendix R
. . .	. . .	
N1 + N2 + 2 + N3	Last pointer for last character as 2nd (back) letter.	
	Spacing information	
N1 + N2 + N3	Length of table	M
+ 3	in records	
+ 4	in words	
+ 5	Default correction value line Pair (none, none)	All = 0
+ 6	Pair (none, 1st back class)	1st correction line for «none» character at front.
+ 7	Pair (none, 2nd back class)	

Word	Meaning	Comment
.	.	
.	.	
.	.	
N1 + N2 + N3 + N5 + 5	Correction value for pair 1st front class, «none» character.	= 0
+6	Correction value for the pair 1st front class, 1st back class	2nd correction line
+7	Correction value for the pair 1st front class, 2nd back class	
. .	.	
. .	.	
. .	.	
2 + N1 + N2 + N3 + M	Correction value for the pair (N5,N5)	Last correction for last front class with last back class.

Correction lines all have a constant length. N4 entries in the letter spacing index usually include empty indicators (indicators to 1st correction line or the first places of the correction line) for character numbers which do not occur in the font in question or which cause no correction, if they are preceding or following a character. In addition, classification sometimes refers to the same place.

Therefore lengths of correction lines are shorter than N4. The following applies:

$M = 2 + N5 \times N5$, where N5 is the number of different classes in a font. In particular N5 is the line length and has e.g., the value 33 (one default correction plus 32 values for 32 different classes).

2. Long kerning record

Word	Meaning	Comment
	Length of long kerning record	2 + N1 + N2 + N3 + M
1	in records	
2	in words	
	Name section	
3	Length of name section	N1 (currently = 3)
4	Number = 3	(long kerning record)
5	Following part number	
	Setting information	
N1 + 3	Length of setting information	N2 (currently = 19)
N1 + 4	Number of lines of correction matrix	N5 (currently)
N1 + 5	Number of columns of correction matrix	N5 (currently)
N1 + 6	Unit size	According to font format
N1 + 7	1st spacing letter	Base upper case
N1 + 8	2nd spacing letter	Base lower case
N1 + 9	Left side bearing	of 1st base spacing letter
N1 + 10	Right side bearing	
N1 + 11	Left side bearing	of 2nd base spacing letter
N1 + 12	Right side bearing	
N1 + 13	1st value of minimum distance	on the level of baselines
N1 + 14	2nd value of minimum distance	between baselines (in touching tables = minimum distance for vertical strokes
N1 + 15	Factor	for compression of white space
N1 + 16	Distance function	= 0 appropriate to geometrical mean = 1 appropriate to arithmetic mean

Word	Meaning	Comment
N1 + 17	Italic angle	in 1/10 degree
N1 + 18	Serif height	in 1/100 mm (are given especially for kerning calculation)
N1 + 19	Cap height	
N1 + 20	x-height	
N1 + 21	Descender length	
	Spacing index	
	Length of spacing	N3 = 2 + N4 × 3
N1 + N2 +3	in records	
N1 + N2 + 4	in words	
N1 + N2 + 5 N1 + N2 + 6	Record and word pointer to correction line for letter pairs in which the 1st (front) letter is character number 1.	The addresses are counted relative to the start of spacing information.
N1 + N2 + 7 N1 + N2 + 8	Record and word pointer to correction line of the letter pairs in which the 1st (front) letter is character number 2.	N4 entries for correction lines .
.	.	≘
.	.	Addresses Z1(i) as per appendix R
.	.	
N1 + N2 + 4 + N4	Word pointer to correction line of the letter pairs in which the last character is in front.	
N1 + N2 + 5 + N4	Word pointer to the place in correction line for letter pairs in which character number 1 is in 2nd (back) position.	N4 entries for correction lines
.	.	≘
.	.	Addresses Z2(j) as per appendix R
.	.	
2 + N1 + N2 + N3	Last word pointer for last character as 2nd (back) letter.	
	Spacing information	
N1 + N2 + N3	Length of tableM	
+ 3	in records	
+ 4	in words	

Word	Meaning	Comment
+ 5	Default correction line first entry for pair (none,none)	All = 0 1st correction line for «none» character at front
+ 6	Pair (none, 1st character)	
+ 7	Pair (none, 2nd character)	
.	.	
.	.	
.	.	
N1 + N2 + N3 + N5 + 5	Default correction value	2nd correction line = 0 (1st char., none)
+ 6	2nd word of correction line for first character with no default value as front letter	Contains correction value for pair formation with first character in front.
+ 7	1st with 2nd character	
+ 8	1st with 3rd character	
. .		
. .		
. .		
2 + N1 + N2 + N3 + M	Last with last character	Last correction line

The correction value for the kerning of character i with its successor j is read from the table as follows:

Address: = Z1(i)+Z2(j) Correction: = contents (address)

kerning

unterschneiden

Fig. 203
Examples of kerning

Touching and Overlapping table

Touching Table

Touching tables do not differ in structure from kerning tables (see Appendix U).

These tables are important in typesetting for two reasons: First, one can find correction values for actual touching, and also gain additional information to help with the kerning of large cap height characters – e.g., in window display text.

It is assumed in outdoor advertising text that kerning works to good effect provided typesetting provides characters with cap heights of up to 10 cm. It should be mentioned, however, that such text lines are generally viewed from a distance of at least 5 meters, which corresponds to viewing a 1 cm character at 50 cm distance.

Larger characters sometimes call for tighter character arrangement. It is as though viewers of large text displays feel that proportionally wider letter separation gives the characters an isolated appearance.

Letter spacing in such cases follows this conception: The user defines in a parameter how close he wishes characters to appear and how near to go to the touching values, with 0% as kerning and 100% just touching. Values between allow appropriate interpolation between kerning and touching.

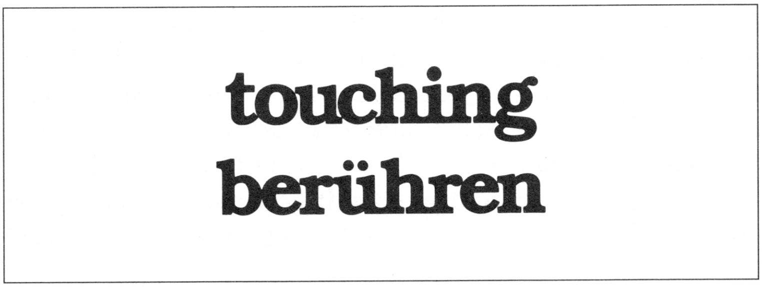


Fig. 204
Examples of touching

Overlapping table

The overlapping tables have the same structure as kerning tables, but contain such large negative character pair correction values that the two letters actually overlap. One application for overlapping is where a white line outlines a character intended to overlap another character. Also one can blend characters into one another, without white outlines. Blending or overlapping of text has absolutely nothing to do with normal text typesetting, and is mainly used in the development of logos and signets or for creating special effects.

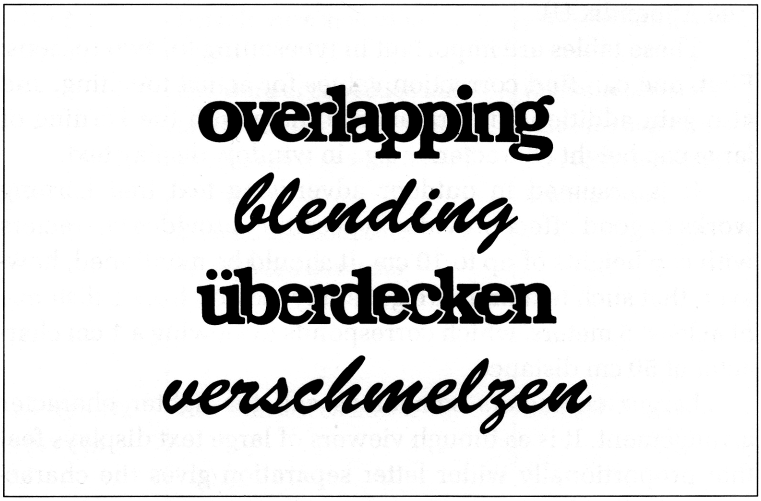


Fig. 205
Examples of overlapping and blending

Programs for converting IK → DI

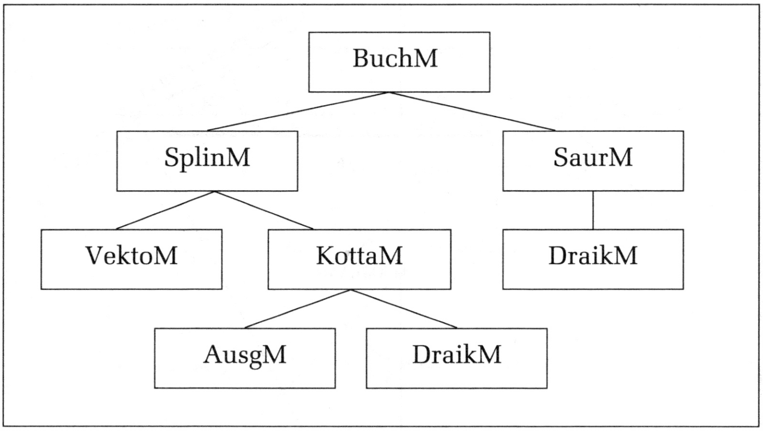
This appendix includes subroutines to convert the IKARUS format into the DISPLAY format.

Everyone is free to copy the C-listings in this chapter. We waive our copyright.

To do this, the function BuchM is called. Prior to this the structure KnotList has to be loaded with the character's data and the variable iktot has to contain the number of digitizations. The result of the computation may be further processed by calling GeradM and KreisM.

For straights the endpoints are stored, for segments of circles endpoints and radii.

The calling sequence is as follows:



The interaction of the functions occurs in such a way that in BuchM the straights and curves are separated from one another. First of all VektoM calculates distances, with related unit vectors, for all IK curve points. The results are stored in the knotlist in DS (distances), C0 (X components) and S0 (Y components). Following this the spline interpolation is started, (see ref. Späth). From the interpolation we take the tangents as unit vectors represented in the form (cos, sin), which are transferred into the knotlist elements C (X components) and S (Y components) after KottaM corrected asymmetries.

The task of DraikM is calculating the sine and cosine of the angles α1 and α2 according to following sketch. Function AusglM is used to perform an adjustment in cases where two circles with a strong difference in radii are joining.

Calculation of the circle segments is based on a method used by the ARISTO company for CAD and executed.

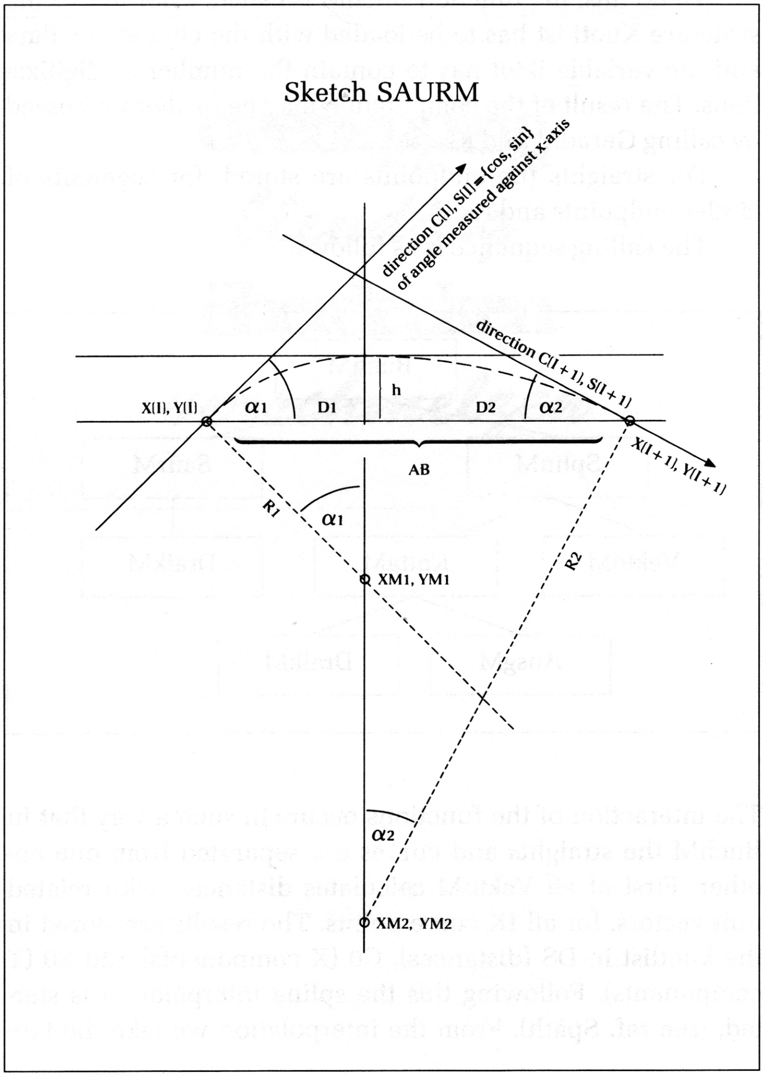

Sketch SAURM
direction C(I), S(I) = (cos, sin)
of angle measured against x-axis
direction C(I + 1), S(I + 1)
h
X(I), Y(I)
α1
D1
D2
α2
X(I + 1), Y(I + 1)
AB
R1
α1
R2
XM1, YM1
α2
XM2, YM2

Fig. 206
The geometrical parameters

Calling the function SaurM. The following sketch are intended to show the method of calculation with which one identifies the two radii R1 and R2 of the circle segments.

Method of calculation

Looking for those two circles which have the given direction in the two points respectively and which join with the direction given by the connection of the two points. One can write down the following:

① $AB = D1 + D2$

② $\sin\alpha_1 = \frac{D1}{R1}$

③ $\sin\alpha_2 = \frac{D2}{R2}$

④ $\cos\alpha_1 = \frac{R1-h}{R1}$

⑤ $\cos\alpha_2 = \frac{R2-h}{R2}$

One has five equations for five unknowns:

D1, D2, R1, R2, h

⑥ → solve ①+②+③:
$AB = R1 \cdot \sin\alpha_1 + R2 \cdot \sin\alpha_2$

④+⑤: $h = R1\,(1-\cos\alpha_1) = R2\,(1-\cos\alpha_2)$

⑦ $$\frac{R1}{R2} = \boxed{DR = \left(\frac{1-\cos\alpha_2}{1-\cos\alpha_1}\right)}$$

⑧ convert 6 : $AB = R2 \cdot \frac{R1}{R2}\sin\alpha_1 + R2 \cdot \sin\alpha_2$

$$= R2\,(DR \cdot \sin\alpha_1 + \sin\alpha_2)$$

$$\Rightarrow \boxed{R2 = AB/(DR \cdot \sin\alpha_1 + \sin\alpha_2)} \Rightarrow \boxed{R1 = DR \cdot R2}$$

Fig. 207

On the following pages we have printed the C - listings in sequence.

```
/*-------------------------------------------------------------------------

   (C) Copyright 1987-92 by URW Unternehmensberatung, Tel.: (040) 606 05 - 0
                                          Harksheider Strasse 102, D-2000 Hamburg 65

 --------------------------------------------------------------------------

   Project:         IKARUS M

   Module:             IKBasic.h

 --------------------------------------------------------------------------

   Description:  BasicIncludeFile for the functions of IKARUS M

 --------------------------------------------------------------------------

   Author:          Date:            Changes:
   Rachow        27.02.1991
   Rachow        06.05.1991       new Knot struct with Float
 ------------------------------------------------------------------------*/

#define    penUp          0
#define    penDown        1

typedef struct                /* this structure describes one IK-point; an array */
{                             /* containing  all points of a character is called KnotList */
   Float  x;                  /* X-coordinate */
   Float  y;                  /* Y-coordinate */
   Short knotType;            /* type of point */
   Float  c;                  /* cosine of tangent */
   Float  s;                  /* sine of tangent */
   Float  g;                  /* auxiliary variable */
   Float  c0;                 /* cosine of cord */
   Float  s0;                 /* sine of cord */
   Float  ds;                 /* distance of two points */
   Boolean selected;          /* auxiliary variable */
}
Knot, *KnotPtr, **KnotHandle;

enum
{
   pntStart = 12,             /* startpoint */
   pntAngle,                  /* cornerpoint  */
   pntCurve,                  /* curvepoint  */
   pntTangt                   /* tangentpoint  */
};
```

```
/*--------------------------------------------------------------------------

   (C) Copyright 1987-92 by URW Unternehmensberatung, Tel.: (040) 606 05 - 0
                                    Harksheider Strasse 102, D-2000 Hamburg 65

 --------------------------------------------------------------------------

   Project:         IKARUS M

   Module:          BuchM.c

   Functions:       BuchM()

 --------------------------------------------------------------------------

   Desription:   Main function for the calculation of lines and arcs for the
                 display of an IK-character. Parts of a contour may be specified.
                 A display factor can be input.

 --------------------------------------------------------------------------

   Author:          Date:            Changes:
   Rachow           22.07.1987
   Knoth            27.11.1987       test if error
   Rachow           28.02.1991       include IKBasic.h
   Rachow           07.05.1991       type Float, HelpKnots removed
   Rachow           22.01.1992       save c,s of knot n, if angle point
 ------------------------------------------------------------------------*/

#include <IKBasic.h>

/*-------------------------------------------------------------------------*/

void BuchM (ianf,iend,istart,iktot,reduceFactor,pknot)

int           ianf;             /* Index of startpoint */
int           iend;             /* Index of endpoint */
int           istart;           /* Index of last startpoint */
int           iktot;            /* # digitizations - 1 */
Float         reduceFactor;     /* display factor */
KnotPtr       pknot;            /* Pointer to the list of IK-points */

/*------------------------------------------------------------------------*/
{
   KnotPtr       pi;
   int        i,m,n,pen;

/* Mark startpoint which is treated like the endpoint of a straight line */

   pen = penUp;
   GeradeM ((pknot + ianf)->x,(pknot + ianf)->y,pen,reduceFactor);

   i = ianf + 1;
   pi = pknot + i;

   while ( i <= iend )                          /* loop handling all poionts */
   {
      if ( pi->knotType != pntCurve )           /* Non-curvepoints are */
      {                                         /* treated as endpoints of straight */
```

```
         if ( pi->knotType == pntStart )         /* lines */
         {
            pen = penUp;
            istart = i;
         }
         else
         {
            pen = penDown;
         }
         GeradeM (pi->x,pi->y,pen,reduceFactor);
      }
      else                                       /* Treatment of curvepoints */
      {
         pen = penDown;
         m = i - 1;

         while ( i < iend )                      /* calculate length of curve */
         {
            i++;
            pi++;

            if ( pi->knotType != pntCurve )
               break;
         }

         if ( pi->knotType < pntAngle      pi->knotType > pntTangt )
         {
            i–;
            pi–;
         }
         n = i;
         if ( n - m <= 1 )                       /* curve is represented by straight line
*/
         {
            GeradeM (pi->x,pi->y,pen,reduceFactor);
         }
         else
         {
            SplinM (pknot,m,n,istart,iktot);     /* calculate tangents in curvepoints*/

            SaurM (m,n,reduceFactor,pknot);      /* calculate arcs */
         }
      }
      i++;
      pi++;
   }
}

/*------------------------------------------------------------------------

   (C) Copyright 1987-92 by URW Unternehmensberatung, Tel.: (040) 606 05 - 0
                                 Harksheider Strasse 102, D-2000 Hamburg 65

 -------------------------------------------------------------------------

   Project:          IKARUS M
   Module:           SplinM.c
   Functions:      . SplinM()

 -------------------------------------------------------------------------
```

```


   Description:     Calculation of the tangents in the curvepoints of an IK-
                    contour. sine and cosine of the slope are stored in the knotlist.
                    The calculation is according to an algorithm using cubic splines
                    described in the book of H. Späth
--------------------------------------------------------------------------------
   Author:          Date:            Changes:
   Rachow           20.07.1987
   Lau              01.09.88         direct reference of sqrt... replaced by math.h
   Rachow           28.02.1991       include IKBasic.h
   Rachow           07.05.1991       type Float, HelpKnots removed
 ---------------------------------------------------------------------------*/

#include <IKBasic.h>
#include <math.h>

/*--------------------------------------------------------------------------*/
void SplinM (pknot,ianf,iend,istart,iktot)
KnotPtr      pknot;           /* Pointer auf die Liste der IK-Punkte */
int          ianf;            /* Index des Anfangspunkts */
int          iend;            /* Index des Endpunkts */
int          istart;          /* Index des letzten Startpunktes */
int          iktot;           /* Gesamtanzahl Digitalisierungen - 1 */
/*--------------------------------------------------------------------------*/
{
   KnotPtr          panf,pend,pn1,pk,pk1;
   int          k,ka,n1;
   Float            ds,z,dxk,dyk,h1,c1,s1,h2,c2,s2;

   /* for all points of the curve calculate slope and distance to the next
      IK-point and store them in the knotlist */

   VektoM (pknot,ianf,iend,istart,iktot);

   n1 = iend - 1;
   pn1 = pknot + n1;

   panf = pknot + ianf;
   pend = pknot + iend;

   switch (panf->knotType)                       /* set the starting values */
   {                                             /* depending on the type of the first */
      case pntTangt:                             /* point */

         ka = ianf + 1;
         panf->g = 0.0;
         panf->c = (panf - 1)->c0;
         panf->s = (panf - 1)->s0;

         break;

      case pntCurve:

         ka = ianf + 1;
         panf->g = 0.0;

         break;

      default:
```

```
      ka = ianf;
      panf->g = 0.5;
      panf->c = 1.5 * panf->c0;               /* set curvature at the end = 0 */
      panf->s = 1.5 * panf->s0;

      break;
}

/* Solving the linear system of equation by using Gauss-Elimination
for tridiagonal systems */

for (k = ianf, pk = panf; k < iend; k++, pk++)
{
   ds = pk->ds;

   if (ds > 0.1)
      h2 = 1.0 / ds;
   else
      h2 = 1.0;

   c2 = 3.0 * pk->c0 * h2;
   s2 = 3.0 * pk->s0 * h2;

   if (k != ianf)
   {
      z = 1.0 / ( 2.0 * (h1 + h2)  -  h1 * pk1->g );

      pk->g = z * h2;
      pk->c = z * (c1 + c2 - h1 * pk1->c);
      pk->s = z * (s1 + s2 - h1 * pk1->s);

      if (k == n1)
      {
         if (pend->knotType == pntTangt)
         {
            pend->c = pend->c0;
            pend->s = pend->s0;

            pk->c  -=  pk->g * pend->c;
            pk->s  -=  pk->g * pend->s;

            break;
         }
         else
         {
            if (pend->knotType == pntCurve  &&  iend != iktot  &&
                                 (pend + 1) -> knotType >= pntAngle)
            {
               pk->c  -=  pk->g * pend->c;
               pk->s  -=  pk->g * pend->s;

               break;
            }
         }
         z = 1.0 / ( 2.0 - pn1->g );

         pend->c = z * ( 3.0 * pn1->c0  -  pn1->c );
         pend->s = z * ( 3.0 * pn1->s0  -  pn1->s );
      }
   }
   c1 = c2;
   s1 = s2;
   h1 = h2;
   pk1 = pk;
}

while (1)
{
   c1 = pk->c;
   s1 = pk->s;

   h2 = sqrt (c1*c1 + s1*s1);

   if (h2 != 0.0)
      h2 = 1.0 / h2;
   else
      h2 = 1.0;

   pk->c = c1 * h2;
   pk->s = s1 * h2;

   k—;
   pk—;

   if (k < ka)
      break;

   pk->c  -=  pk->g * c1;
   pk->s  -=  pk->g * s1;
}

switch (pend->knotType)                      /* special treatment of the endpoint */
{
   case pntCurve:

      if ( pend->x != panf->x       pend->y != panf->y )
         break;

      dxk = panf->c + pend->c;
      dyk = panf->s + pend->s;

      h2 = sqrt (dxk*dxk + dyk*dyk);

      if (h2 == 0.0)
         break;

      h2 = 1.0 / h2;

      panf->c = dxk * h2;
      panf->s = dyk * h2;

      pend->c = panf->c;
      pend->s = panf->s;

      break;

   case pntAngle:

      (pend - 2)->g = pend->c;
      pn1->g = pend->s;
```

```
          if ( iend != iktot  &&  (pend + 1)->knotType >= pntAngle )
             break;

          if ( pend->x != panf->x      pend->y != panf->y )
             break;

          pend->c = (pknot + istart)->c;
          pend->s = (pknot + istart)->s;
   }
   KottaM (pknot,ianf,iend);                      /* modify tangents if necessary */
}

/*-------------------------------------------------------------------------------
   (C) Copyright 1987-92 by URW Unternehmensberatung, Tel.: (040) 606 05 - 0
                                    Harksheider Strasse 102, D-2000 Hamburg 65
 -------------------------------------------------------------------------------
   Project:          IKARUS M
   Module:           VektoM.c

   Functions:     1) VektoMCalc()
                  2) VektoM()
 -------------------------------------------------------------------------------
   Description:   calculation of the slope of the secant and the distance to the
                     next IK-point of a contour and storage in the knotlist
 -------------------------------------------------------------------------------
   Author:           Date:             Changes:
   Rachow         17.07.1987
   Lau            01.09.1988   direct reference of sqrt..replaced by math.h
   H.Schwarz      04.01.1990   static Vekto4Calc()
   Rachow         28.02.1991   include IKBasic.h
   Rachow         07.05.1991   type Float, HelpKnots removed
 -------------------------------------------------------------------------------*/

#include <IKBasic.h>
#include <math.h>

static void VektoMCalc (pknot, x, y)

KnotPtr          pknot;       /* pointer to the list of IK-points */
Float            x,y;         /* coordinates of the next IK-point */

/*-------------------------------------------------------------------------------*/

{
   Float dx,dy,ds;

   dx = x - pknot->x;
   dy = y - pknot->y;

   ds = sqrt (dx*dx + dy*dy);

   pknot->ds = ds;

   if (ds == 0.0)
   {
      pknot->c0 = 1.0;
      pknot->s0 = 0.0;
   }
   else
   {
      pknot->c0 = dx/ds;
      pknot->s0 = dy/ds;
   }
}
```

```
/*-----------------------------------------------------------------------------*/

void VektoM (pknot,ianf,iend,istart,iktot)

KnotPtr       pknot;          /* pointer to the list of IK-points */
int           ianf;           /* index of startpoint */
int           iend;           /* index of endpoint */
int           istart;         /* index of last startpoint */
int           iktot;          /* # digitizations - 1 */
/*-----------------------------------------------------------------------------*/
{
   KnotPtr       pend,pi,pi1;

   pend = pknot + iend;

   if ((pknot+ianf)->knotType == pntTangt ) /* Anfang tangential */
      ianf—;

   for (pi = pknot+ianf; pi < pend; pi++)
   {
      pi1 = pi + 1;
      VektoMCalc (pi,pi1->x,pi1->y);
   }

   if ( pend->knotType == pntTangt )           /* Ende tangential */
   {
      if ( iend == iktot    (pend+1)->knotType < pntAngle )
         pi1 = pknot + istart + 1;
      else
         pi1 = pend + 1;

      VektoMCalc (pend,pi1->x,pi1->y);
   }
}


/*-----------------------------------------------------------------------------

   (C) Copyright 1987-92 by URW Unternehmensberatung, Tel.: (040) 606 05 - 0
                                  Harksheider Strasse 102, D-2000 Hamburg 65

 ------------------------------------------------------------------------------

   Project:          IKARUS M

   Module:           KottaM.c

   Functions:        KottaM()

 ------------------------------------------------------------------------------
```

```
   Description:  Correction of tangents
 -------------------------------------------------------------------------------

   Author:          Date:            Changes:
   Rachow           21.07.1987
   Rachow           28.02.1991       include IKBasic.h
   Rachow           08.05.1991       type Float, HelpKnots removed
 -----------------------------------------------------------------------------*/

#include <IKBasic.h>

/*----------------------------------------------------------------------------*/

void KottaM (pknot,m,n)

KnotPtr      pknot;           /* pointer to list of IK-points*/
int          m;               /* index of startpoint */
int          n;               /* index of endpoint */

/*----------------------------------------------------------------------------*/
{
   KnotPtr       pm,pm1,pn,pi;
   int           in,im;
   Float         c1,s1,c2,s2,c3,s3,cs,ss,css,sss,sa1,sa2,sa11,sa22,ca1,ca2,ca11,
                 ca22;

   pm  = pknot + m;
   pm1 = pm + 1;
   pn  = pknot + n;

   c1 = pm->c;
   s1 = pm->s;

   c2 = pm1->c;
   s2 = pm1->s;

   cs = pm->c0;
   ss = pm->s0;

   in = DraikM (cs, ss, c1, s1, c2, s2, &sa1, &sa2, &ca1, &ca2);

   for (pi = pm1; pi < pn; pi++)
   {
      if ( (pi + 1)->knotType  !=  pntAngle )
      {
         c3 = (pi + 1)->c;
         s3 = (pi + 1)->s;
      }
      else
      {
         c3 = (pi - 1)->g;
         s3 = pi->g;
      }

      css = pi->c0;
      sss = pi->s0;
```

```
        im = DraikM (css, sss, c2, s2, c3, s3, &sa11, &sa22, &ca11, &ca22);

        if (in != 0)
        {
            if (im != 0)
            {
                if (in * im >= 0)
                {
                    if (AusglM (in, sa1, sa2, sa11, sa22, &c2, &s2)  !=  0)
                    {
                        im = DraikM (css, sss, c2, s2, c3, s3, &sa11, &sa22,
                                                          &ca11, &ca22);

                        pi->c = c2;
                        pi->s = s2;
                    }
                }
            }
            else
            {
                c2 = css;
                s2 = sss;

                c3 = css;
                s3 = sss;

                pi->c = c2;
                pi->s = s2;

                (pi + 1)->c = c3;
                (pi + 1)->s = s3;
            }
        }

        in = im;

        sa1 = sa11;
        sa2 = sa22;

        ca1 = ca11;
        ca2 = ca22;

        cs = css;
        ss = sss;

        c1 = c2;
        s1 = s2;

        c2 = c3;
        s2 = s3;
    }

    pi->c = c2;
    pi->s = s2;
    if (pi->knotType  !=  pntCurve)
    {
        (pi - 2)->g = c2;
        (pi - 1)->g = s2;
    }
}
```

```
/*---------------------------------------------------------------------------
   (C) Copyright 1987-92 by URW Unternehmensberatung, Tel.: (040) 606 05 - 0
                                 Harksheider Strasse 102, D-2000 Hamburg 65
 ---------------------------------------------------------------------------

   Project:         IKARUS M

   Module:          DraikM.c

   Functions:       DraikM()

 ---------------------------------------------------------------------------

   Description:     Calculates the angles in a rightangled triangle which is
                    spanned by the cord between two IK-points and the tangents in
                    these points.

 ---------------------------------------------------------------------------

   Author:      Date:        Changes:
   Rachow       21.07.1987
   Rachow       27.02.1991   include IKBasic.h
   Rachow       08.05.1991   type Float, HelpKnots removed
 -------------------------------------------------------------------------*/

#include <IKBasic.h>

/*-------------------------------------------------------------------------*/

int DraikM (cs, ss, cil, sil, ci, si, psa1, psa2, pca1, pca2)

Float           cs,ss;          /* sin/cos of slope of cord */
Float           cil,sil;        /* sin/cos of slope of tangent in 1st IK-point */
Float           ci,si;          /* sin/cos of slope of tangent in 2nd IK-point */
Float           *psa1,*pca1;    /* pointer to sin/cos of 1st angle */
Float           *psa2,*pca2;    /* pointer to sin/cos of 2nd angle */
/*-------------------------------------------------------------------------*/
{
   Float         sa1,sa2;
   int        in;

   sa1 = cs * sil - cil * ss;
   sa2 = cs * si  - ci  * ss;

   *psa1 = sa1;
   *psa2 = sa2;

   if (sa1 > 0)
   {
      if (sa2 > 0)
         return 0;

      sa2 = - sa2;
      *psa2 = sa2;

      in = -1;
   }
   else
   {
      if (sa2 < 0)
         return 0;

      sa1 = - sa1;
      *psa1 = sa1;

      in = 1;
   }

   if (sa1 < 0.0015      sa2 < 0.0015)
      return 0;

   *pca1 = cs * ci1 + ss * si1;
   *pca2 = cs * ci  + ss * si;

   if (*pca1 < 0.1      *pca2 < 0.1)
      return 0;

   return in;
}

/*------------------------------------------------------------------------------
    (C) Copyright 1987-92 by URW Unternehmensberatung, Tel.: (040) 606 05 - 0
                                    Harksheider Strasse 102, D-2000 Hamburg 65
 -------------------------------------------------------------------------------
   Project:         IKARUS M
   Module:             AusglM.c
   Functions:    AusglM()
 -------------------------------------------------------------------------------
   Description:  Compensation of triangles which are too skewed. One
   cathetus of each triangle forms a straight line. This
   common direction is corrected to obtain more symmetrical
triangles.
 -------------------------------------------------------------------------------
   Author:          Date:            Changes:
   Rachow        21.07.1987
   Lau           01.09.88         direct referencing of sqrt replaced by math.h
   Schwarz       23.02.89         if(cdels<0.99999) removed
   Rachow        27.02.1991       include IKBasic.h
   Rachow        07.05.1991       type Float, HelpKnots removed
 -----------------------------------------------------------------------------*/

#include <IKBasic.h>
#include <math.h>

/*----------------------------------------------------------------------------*/

int AusglM (idreh, sa1, sa2, sa11, sa22, pci, psi)

int       idreh;                  /* sense of rotation of the two triangles */
Float     sa1,sa2;                /* sin of angles of 1st triangle */
Float     sa11,sa22;              /* sin of angles of 2nd triangle */
Float     *pci,*psi;              /* input: common direction of cathetus*/
                                  /* output: new common direction */

                                  /* return value: */
                                  /* =0: no correction */
                                  /* =1: correction executed */
/*----------------------------------------------------------------------------*/
{
    Float dr,del,del1,del2,deldr,c,cdel,cdels;

    dr = idreh * 0.7;

    del1 = (sa1  - sa2 ) * dr;
    del2 = (sa11 - sa22) * dr;


    del = (del1 + del2) * 0.7;
    deldr = del * dr;

    if (deldr >= 0)                                 /* del positive: rotation right */
    {                                               /* del negative: rotation left */
        if (deldr < 0.01)
            return 0;

        if (deldr > del2 * dr)
            del = del2;                             /* sa11 gets smaller */
    }
    else
    {
        if (deldr > -0.01)
            return 0;

        if (deldr < del1 * dr)
            del = del1;                             /* sa2 gets smaller */
    }

    cdels = 1.0 -  del * del;

    cdel = sqrt (cdels);                            /* rotate (ci,si) */

    c = (*pci) * cdel - (*psi) * del;

    *psi = (*pci) * del + (*psi) * cdel;
    *pci = c;

    return 1;
}
```

```
/*------------------------------------------------------------------------------
   (C) Copyright 1987-92 by URW Unternehmensberatung, Tel.: (040) 606 05 - 0
                                    Harksheider Strasse 102, D-2000 Hamburg 65
 -------------------------------------------------------------------------------
   Project:         IKARUS M
   Module:          SaurM.c

   Functions:       SaurM()
 -------------------------------------------------------------------------------
   Description:     Calculates two arcs between two curvepoints according to
                    the SAUR-algorithm. In the IK-points the arcs get the tangents which
                    were calculated with splinM, in the meeting-point between the
                    IK-points  the get the slope of the cord.
  -------------------------------------------------------------------------------
   Author:          Date:            Changes:
   Knoth            20.08.1987       where it all began
   Knoth            29.10.87         dividing the struct Knot
   Rachow           28.02.1991       parameter change
   Rachow           07.05.1991       type Float, HelpKnots removed
 ---------------------------------------------------------------------------*/

#include <IKBasic.h>

/*---------------------------------------------------------------------------*/
void SaurM(m,n,reduceFactor,knotList)

int              m;               /* index of startpoint */
int              n;               /* index of endpoint */
Float            reduceFactor;    /* display factor */
KnotPtr          knotList;        /* pointer to list of IK-points */

/*---------------------------------------------------------------------------*/
{
   int    nn1, i, j, in;

   Float  dr, ca, ce, r1, r2, sh, xm1, ym1,
          xm2, ym2, xh1, yh1, xa, ya,sa,se;

   Boolean   kurve;

   nn1 =  penDown;

   /*---------------loop handling all IK-points--------------------*/

   for (i=m+1;   i<= n; i++)
      {
      kurve = true;
      j = i  - 1;   /* index of previous point*/

      /* calculate parameters of angle */

      in = Draik4((knotList+j)->c0,(knotList+j)->s0,(knotList+j)->c,
                 (knotList+j)->s,(knotList+i)->c,(knotList+i)->s,&sa,&se,&ca,&ce);

      if (in==0) kurve = false;          /* if it is a small angle, draw a straight line*/
         else
         {
```

```
    /* else calculate arcs */

        dr  = (1.-ce)/(1.-ca);
        r2  = (knotList+j)->ds/(dr*sa+se);    /* radius of 2nd circle*/
        sh  = r2*(1.-ce);                     /* cordheight */

        if  (sh < reduceFactor)
            kurve = false;                    /* if cordheight is small, straight line*/
        else
            {
            r1  = dr*r2;

            if  ( (r1 < 1e-5) && (r1 > -1e-5))
                kurve = false;                /* small radius, straight line*/
            else
                {
                if  (in<0)                    /* only positive radii*/
                    {
                    r1  = -r1;
                    r2  = -r2;
                    }

        /* calculate center points and auxiliary points */

                xm1  = (knotList+j)->x - (r1 * (knotList+j)->s);
                ym1  = (knotList+j)->y + (r1 * (knotList+j)->c);

                xm2  = (knotList+i)->x - (r2 * (knotList+i)->s);
                ym2  = (knotList+i)->y + (r2 * (knotList+i)->c);

                xh1 = xm1 + (r1 * (knotList+j)->s0);
                yh1 = ym1 - (r1 * (knotList+j)->c0);

                xa  = (knotList+j)->x;
                ya  = (knotList+j)->y;


        /* draw first arc */

                KreisM (xa,ya,xh1,yh1,xm1,ym1,r1,sh,reduceFactor);


                xa  =  xh1;
                ya  =  yh1;

        /* draw second arc */

                KreisM (xa,ya,(knotList+i)->x,(knotList+i)->y,xm2,ym2,r2,sh,
                                                      reduceFactor);
                }
            }
        }
    if (kurve==false)
        GeradeM ((knotList+i)->x,(knotList+i)->y,nn1,reduceFactor);
    }
}
```

Appendix X

Marking and digitizing rules

Marking

Prior to the proper digitizing procedure, an original must be prepared. This task involves marking digitization points, adjusting the baseline, left and right side bearings, and treating of special cases, e.g., digitizing multi-colored logotypes.

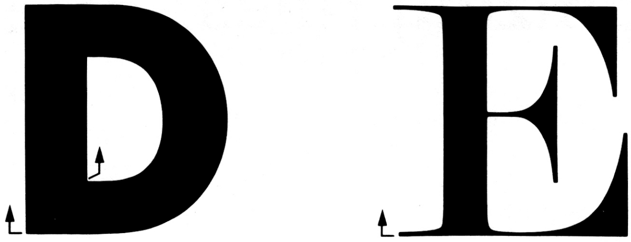
First mark, then digitize. Never both at the same time.

The results of digitizing are very greatly dependent upon the number and position of digitization points. The ideal is achieving the best reproduction of an original with the least number of points. To this end it is recommended marking points to be digitized on the original copy prior to digitization, as follows:

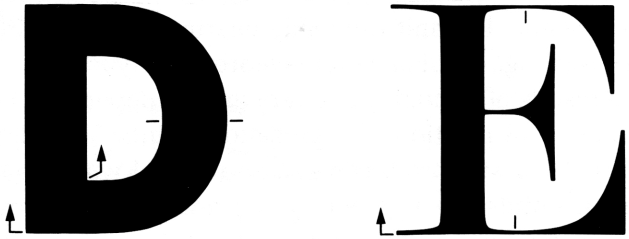
Start points	↴	Show the direction of digitization.
Corner points	( )	Need not be marked, they are obvious.
Curve points	(–)	Extreme points, two or three points within a 90° change of direction, at points of inflection and auxiliary points.
Tangent points	(<)	Tangent points never occur singly. There are either two tangent points or a neighbouring corner point, in order to define the straight.

Initially, all the necessary points (e.g., curve extremes and auxiliary points) are marked, followed by the necessary intermediate points, marked as uniformly as possible.

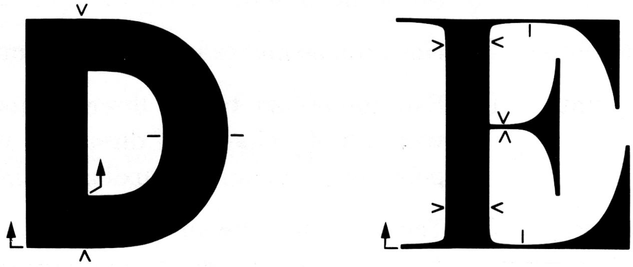
1) Mark in start points



2) Mark in curve extreme points



3) Mark in tangent points



4) Mark in aid points

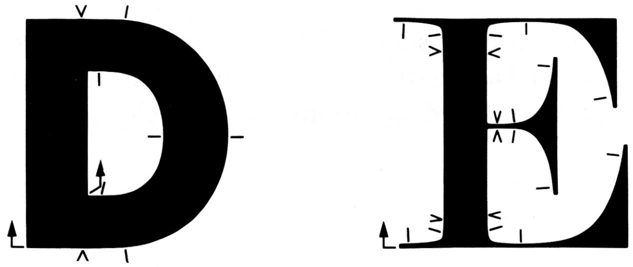


Fig. 208
Digitizing rules
for start, corner
and tangent points

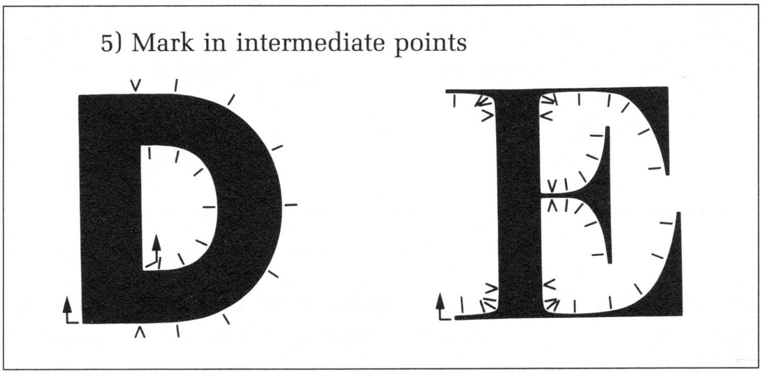


Fig. 209
Rules for curve points

In the following, further marking rules are described using combinations of text and examples.

A circle is marked by its extreme points. These points are located at the maximum and minimum extension of a curve. With a full circle, four curve points suffice for its display. Deviations from ideal circles require additional curve points; two or three additional curve points per 90 degree turn.

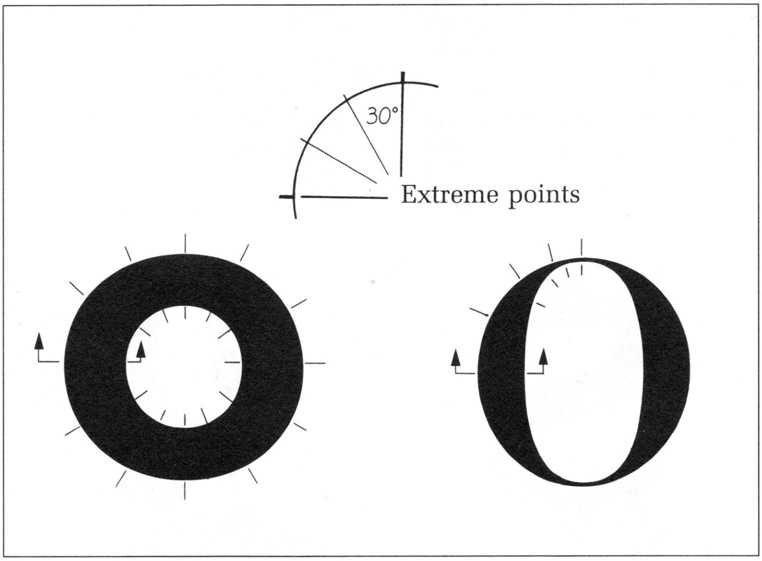


Fig. 210
A curve point ever 30°

The join of a curve to a corner requires special attention. As only curve points can create curved lines, placing a curved point too far from the join produces too flat a curve.

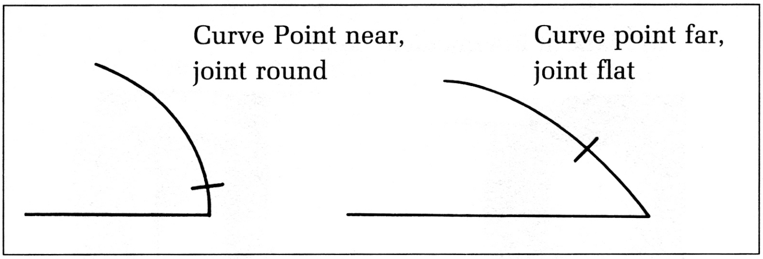


Fig. 211
Some curve points are round, others are angular.

To avoid this, the final curve point must be located approximately three to five millimeters from a corner point (assuming 100 mm cap height).

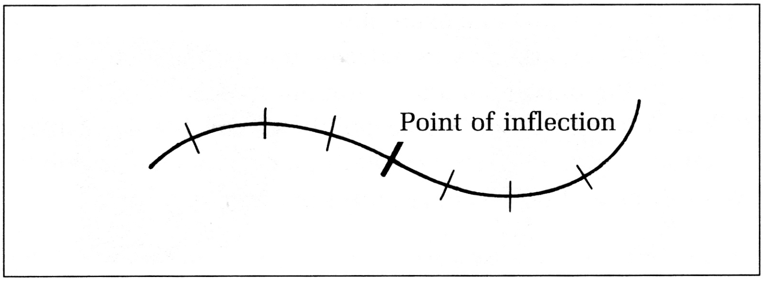


Fig. 212
Marking a point of inflection

Points of inflection, referring to those positions at which a right curve runs into a left curve or a left curve runs into a right curve, must be marked with a curve point.

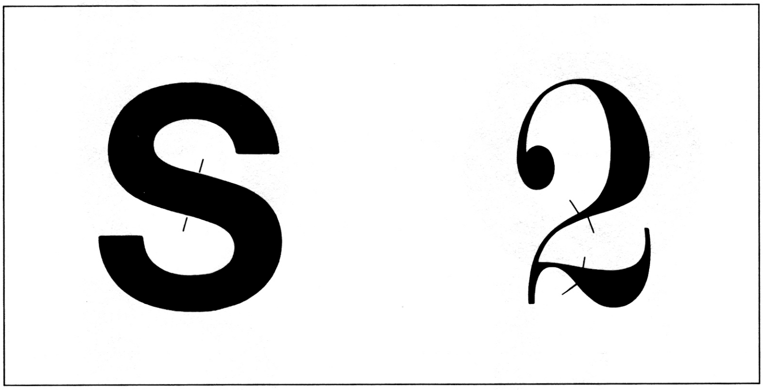


Fig. 213
Position of points of inflection

There are instances in graphics where such directional changes take place gradually, where a straight segment occurs between two curved sections. In these cases we do not place a point of inflection, rather a transition from curve into straight and from straight into curve, i.e. tangent points.

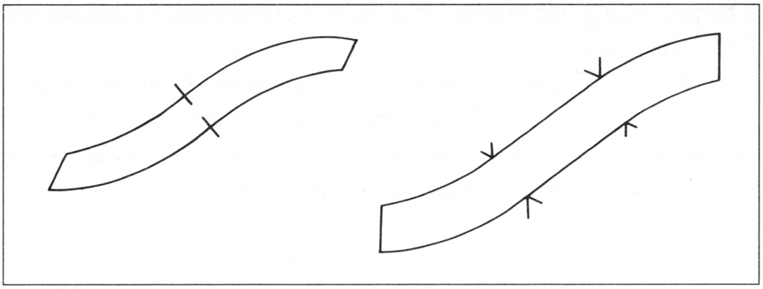


Fig. 214
Point of inflection with «straight» turn

Marking examples

The original shown below has only straight contours. Correspondingly we have one start and twelve corner points to digitize. The direction of digitization is marked at the start point with an arrow. The corner points need not be marked.

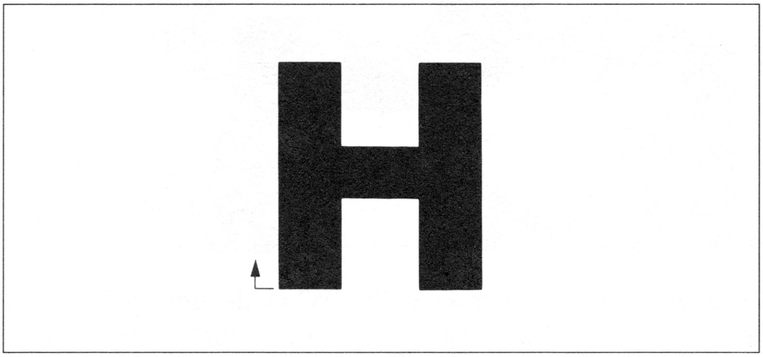


Fig. 215
One start point per contour

The original below-right has only tangent and curve points. For the start point, the correct tangent point is selected so the second point will also be a tangent point.

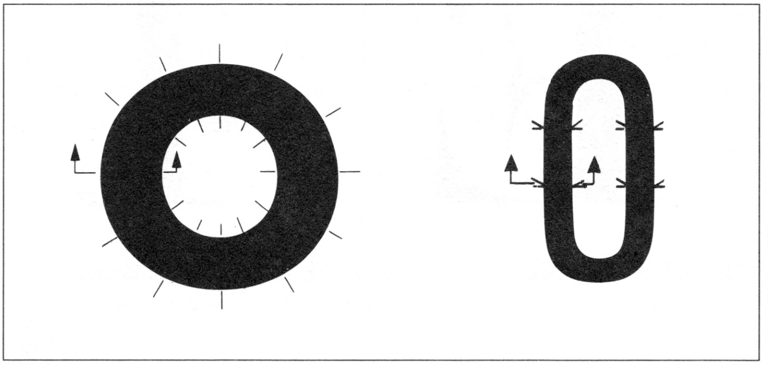


Fig. 216
If there are no corners, position the start point on one of the extrema or at the start of a straight.

The preceding left original has only curve points. Points should be marked approximately every 30 degrees.

Original below consists of corner and curve points. Notice that extreme values of curves must be marked, and that at the section joining corner to curve, a curve point should be set approximately 3 to 5mm from the corner.

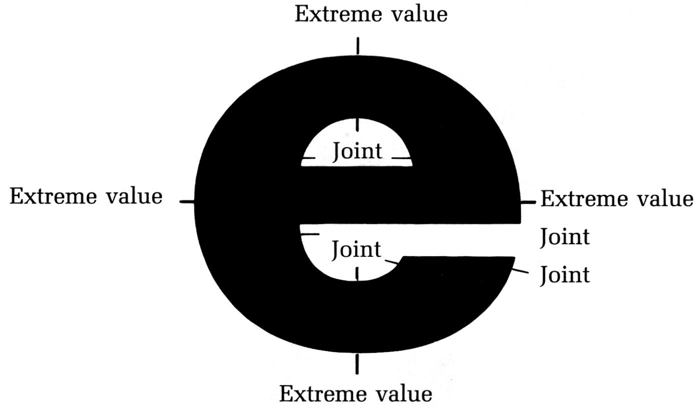

Extreme value
Joint
Extreme value
Extreme value
Joint
Joint
Joint
Extreme value

Fig. 217
Positioning of curve points at corner

Tangent points are marked with >. When dealing with symmetrical sections of an original, be certain that corresponding tangent points are marked at the same height or width.

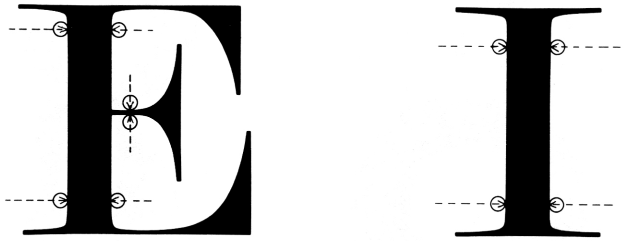


Fig. 218
Positioning of tangent points

Digitizing rules

1. Operating the sensor

The digitizing sensor utilizes a cross-hair fitted under a magnifying glass and five sensor keys. These are fitted into a container and the unit interacts electromagnetically with the digitizer tablet. This small device is popularly known as a «mouse». In order to avoid misunderstanding, it should be pointed out that the commonly used computer mouse simply returns relative displacement information to its host computer, whereas a digitizer mouse enables absolute measurements of positions.

It is possible, to digitize a point every two seconds. But, if the point ends up in the wrong place, it takes over 20 seconds to correct it interactively. So:
1. slow and steady
+
2. concentration
=
more speed in the long run.

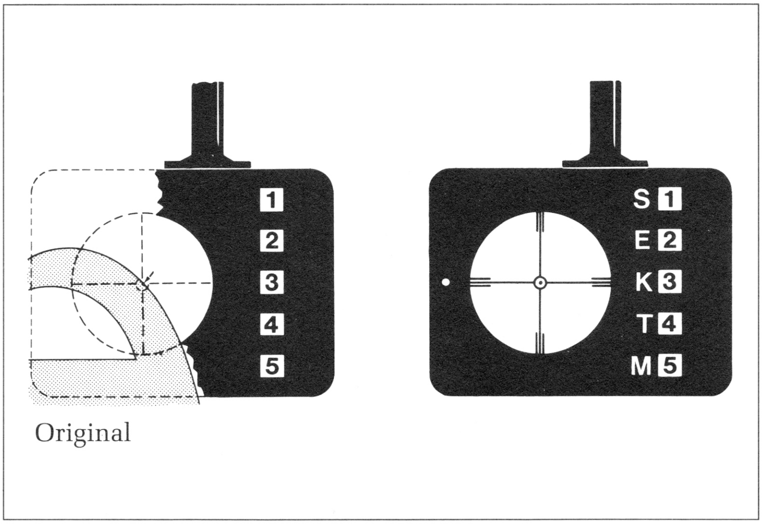


Fig. 219
Start - corner - curve - tangent is the digitizing sequence.
S C C T
In German we say: Start - Ecke - Kurve - Tangente, in short
S E K T,
the German word for sparkling wine.

Cross-hair

The point where the two hairs cross is aligned with a marked point on the surface of the black/white rendition of the letter outline. Its current position is input to the computer by pressing the appropriate key for the type of point desired.

Sensor keys

The upper four sensor keys correspond directly to the four individual types of points used to represent the IKARUS format:

s	**1**	start point	**s**
e	**2**	corner point	**c**
k	**3**	curve point	**c**
t	**4**	tangent point	**t**

The last sensor key

m	**5**	is reserved for the menu field

and initiates the menu command.

To select from the menu the cross-hair is placed close to the center of the relevant 1 cm × 1 cm menu box (lining-up precisely with the center of the box is not required) and sensor key m 5 pressed.

The menu commands allow 34 different program functions to be selected for both new digitizing and for correction of digitizations on originals.

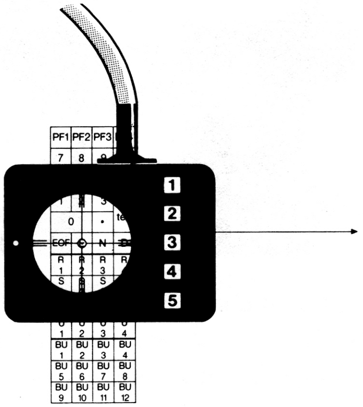

PF1
PF2
PF3
7
8
0
EOF
N
R 1 S
R 3 S
1
2
3
4
5
BU 1
BU 2
BU 3
BU 4
BU 5
BU 6
BU 7
BU 8
BU 9
BU 10
BU 11
BU 12

Fig. 220
Menu and five-button digitizer

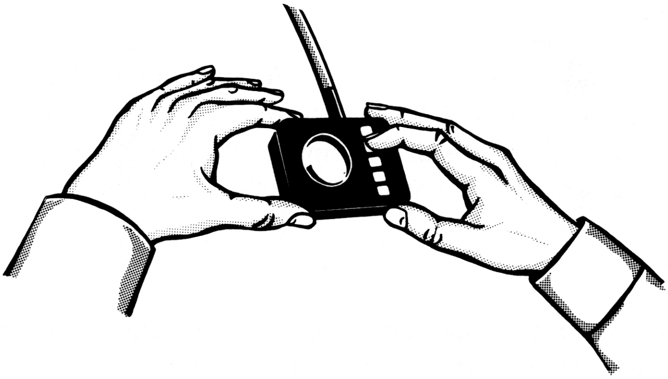


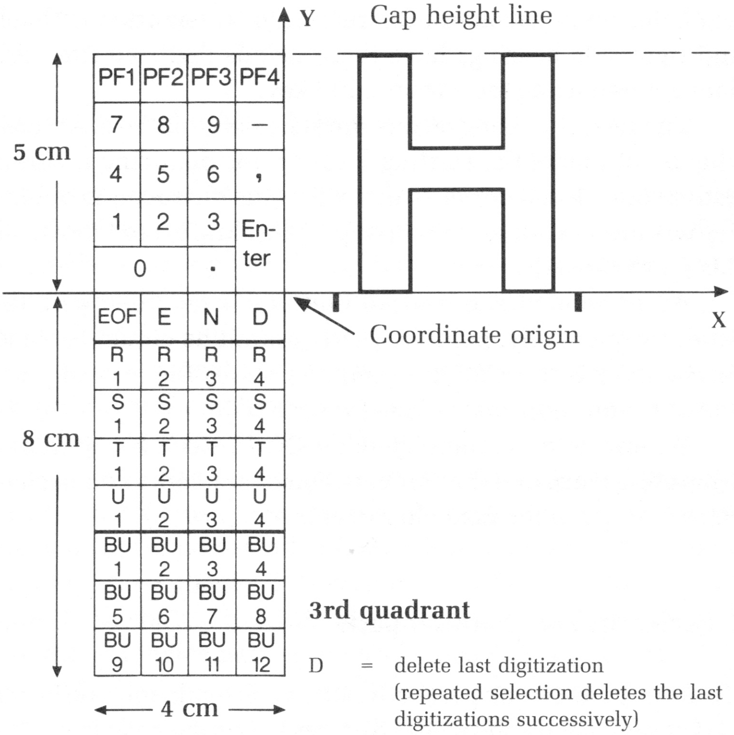
Fig. 221
You can move the digitizer more steadily by holding it with both hands.

2. Structure of digitizing menu for IKARUS

The axis intersection together with the menu box is marked on a transparent sheet placed on the digitizer tablet.

The menu field in the 2nd quadrant can be used as an alternative to the key board when entering character numbers.

The other fields are used with an interactive program not described here.



PF1	PF2	PF3	PF4
7	8	9	–
4	5	6	,
1	2	3	Enter
0		.	
EOF	E	N	D
R 1	R 2	R 3	R 4
S 1	S 2	S 3	S 4
T 1	T 2	T 3	T 4
U 1	U 2	U 3	U 4
BU 1	BU 2	BU 3	BU 4
BU 5	BU 6	BU 7	BU 8
BU 9	BU 10	BU 11	BU 12

3rd quadrant

D = delete last digitization (repeated selection deletes the last digitizations successively)
N = next character (≙ end of character mark)
E = end of a font
EOF = end of all digitizations
R = R serif of e.g.
S = S serif
T = T serif
U = U serif
BU1 …BU12 = character parts of numbers 1 to 12

Fig. 222
A typical menu field on the digitizer tablet

3. Digitizing

A character original is fixed with adhesive tape or on register pins under the plastic sheet so its baseline rests exactly on the X axis and its left side bearing about 1 cm to the right of the Y axis. No part of the character may overlap the menu field at the left of the Y axis.

Unlike preliminary marking, digitizing of all characters must take place in a prescribed sequence.

Beginning the digitizing procedure the coordinate origin for the entire font is determined, simultaneously defining the position of the IKARUS menu field. A job number is entered using the menu keyboard. Characters to be recorded with left and right side bearings have these details digitized first, this data entered using the corner point key.

The next digitizing step is the start point of the character. This point marks the starting location for digitizing a closed outline or a closed curve so that when the character is output, the writing head (pen, engraving stylus, knife, etc.) moves to this point «raised».

As mentioned, the contour (outline) is recorded with the sensor sequentially, point by point around its circumference. As each key is pressed, the computer stores the type of point and absolute coordinate value for each point.

When a letter is completed, an «N» (next) is keyed in as a separating mark or delimiter between characters and the character number of the next character is entered.

4. Part-serifs and character parts

Up to four different part-serif sets, each with four different part-serifs and up to twelve different character parts can be stored by processing steps.

Storing part-serifs

One needs to digitize only the R1 or S1, T1 or U1 serif, paying attention that one always «digitizes from the serif to the straight», because the program always assumes that all parts of serifs are digitized as if they were positioned on vertical stems. Prior to the recording of part-serifs the appropriate menu command must be given. The program calculates the part-serifs 2, 3 and 4 automatically as mirror images at Y axis, at X axis and as rotation by 180 degrees, and stores them ready for recall via other menu commands.

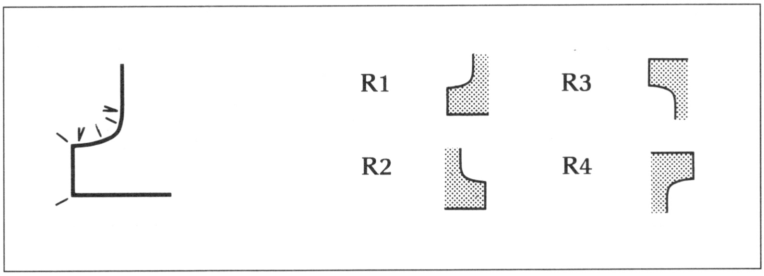


Fig. 223
Part-serifs are mirrored automatically.

Recording and fitting of character parts

The term «character parts» refers to parts similar to serifs (incomplete, characteristic letter forms, repeated regularly throughout an alphabet), and has relevant commands in the menu positioned directly under those given for serifs. The procedure for digitizing character parts differs from that of serifs in so far that after the menu command is given for the recording of the character parts, they start with a start point and they are digitized «from the straight to the serif».

In contrast to serifs, character parts must be digitized at the correct Y height. One must make certain that the last point (here a corner point) in the X direction allows the direction of fitting to be recognized. Directions of fitting are «to the right» and «to the left». This shift takes place only in the X direction, i.e. the character part must be digitized for recording at the correct Y height. Prior to digitizing the character part, the appropriate menu command (BU1 to BU12) must be given.

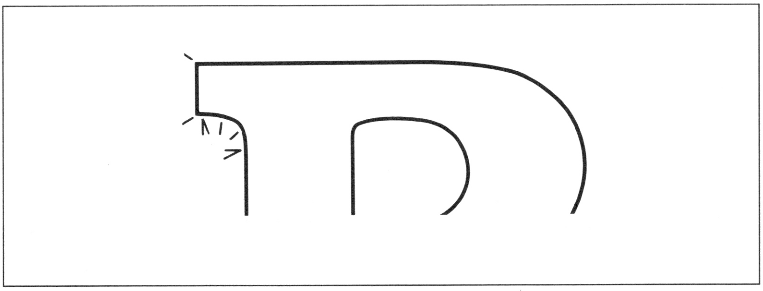


Fig. 224
Character parts can be recalled automatically.

Digitizing part-serifs

The digitizing sequence is: Start point, tangent point, menu field R2, corner point, corner point, menu field R1, tangent point, tangent point, menu field R3, corner point, corner point, menu field R4, tangent point. Care must be taken to ensure that digitizing a character does not start or end with a menu command. In the course of further processing (channel processing), the two horizontal straights of the «I» (for example) are aligned with the base line and cap height line respectively. Later, serifs are fitted using the program DV.

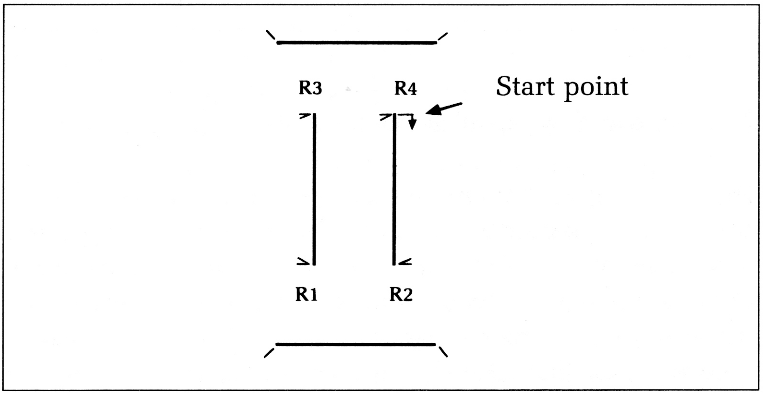


Fig. 225
Characters are digitized without part-serifs.

Serif fitting on straight stems

Part-serif a is to be fitted onto character b. Points 5 and 6 are determinant for the length of the serifs, points 3 and 4 are not determinant with respect to their Y position, which is taken from part a (serif height is defined by point 2). The curved segment of the serif is retained, the extension lies between corner and tangent point. The result is character part c.

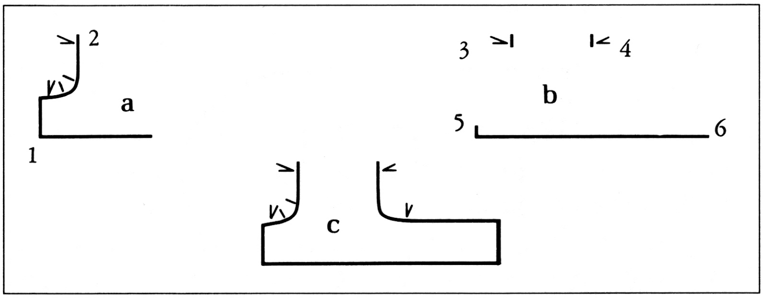


Fig. 226
Integration of half and full serifs

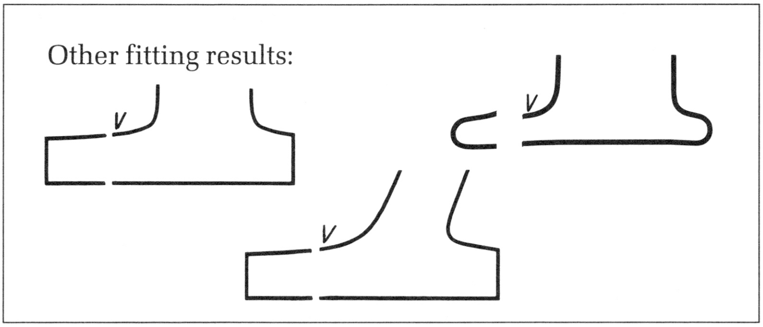


Fig. 227
Fitting serif extrema

As is seen in this example, it is imperative that serifs to be fitted are marked with tangent points.

Serif fitting on sloped strokes

Serifs may be fitted not only to vertical but also to sloped stems. In this case the curve of the serif has to be adjusted to match the diagonal. For this, so-called «alignment points» are required to detail the thickness and direction of the sloping stems.

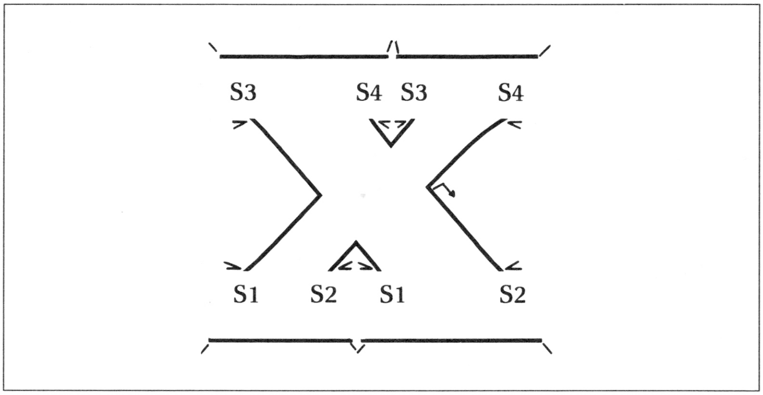


Fig. 228
Fitting serifs on sloped strokes

All points in this example also serve as alignment points. In the case of fitting onto straight sloping stems, two alignment points would be sufficient for the program.

When the alignment points are curve points, round fitting is executed. Three alignment points will always be required to determine and describe a circle. The last point of the serif part is placed on the circle at its correct Y height. Finally the alignment point which is nearest to the serif is deleted.

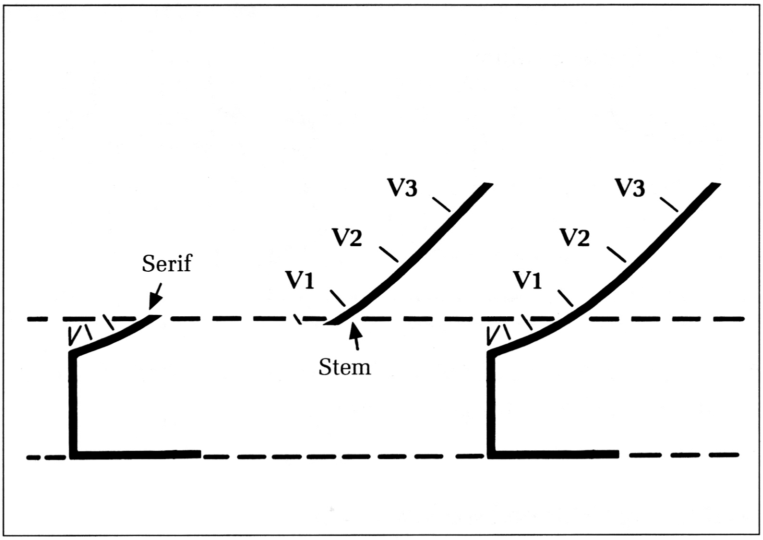


Fig. 229
Curves are automatically aligned when serifs are fitted.

5. Post digitizing

When only slight corrections to letters must be executed, it is possible to post-digitize segments rather than carry out a completely new digitization. To match digitizations of the first character, the correct character number is entered and the left and right side bearings digitized to act as calibration points. The first and last point of the newly recorded part are fitted to the character. A tolerance circle at the start and end points of the post digitization allows automatic fitting of the respective points to the corrected digitization.

First and last points of the post digitization are deleted. Closed lines can also be similarly fitted. A tolerance circle around the start point allows the appropriate closed line to be found.

The number and types of points digitized in the old and new parts do not have to correspond with one another, and the number of points is recalculated.

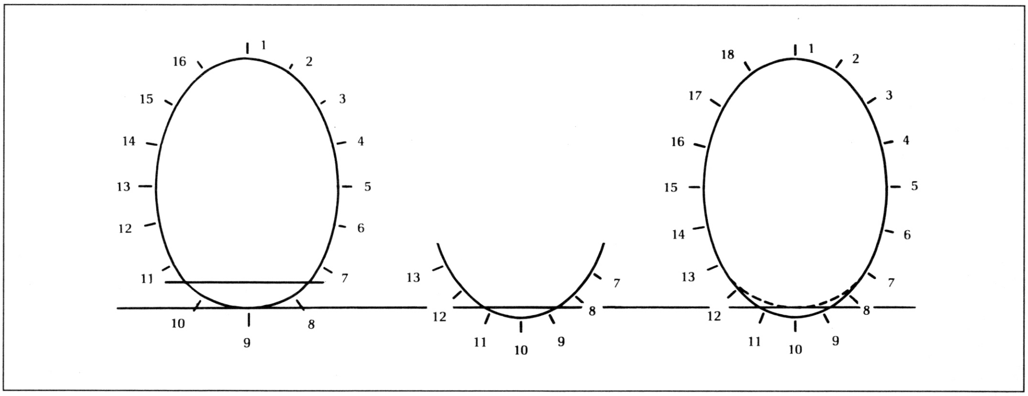


Fig. 230
Points are moved; curves are automatically realigned.

6. Experience

You can explain the digitizing rules to every new colleague in about 15 minutes. Nevertheless, it takes about three month to get acqainted with hand-digitizing in a team and to work without leaving much to the quality assurance people. But it takes three years to be three times as fast as after three months.

You can compare it very well with the work of a bricklayer in house building. Also there, a master can explain how to do it in about 15 minutes: how to use the trowel, how to mix the concrete mortar and take it with the trowel and put it on a brick, and how to place a brick onto brickes.

Afterwards a new colleague can't built all walls for a house straight and steady after three months, but doubtless after three years.

References, on typefaces in particular

1. Typefaces

Bosshard, H. R., «Technische Grundlagen zur Satzherstellung» Band 1, Verlag des Bildungsverbandes Schweizerischer Typografen Bst, Bern, 1980.

Fiebig, D., «Tabellensatz. Programmierte Anleitung für das richtige Setzen von Reihensatz, Formularsatz, Werk-, Akzidenz- und Schreibtabellen», Heft F 11 Fachtechnische Schriftenreihe, Industriegewerkschaft Druck und Papier, Stuttgart, 1971.

Frutiger, A., «Schrift – Ecriture – Lettering. Die Entwicklung der europäischen Schriften, in Holz geschnitten», Verlag des Bildungsverbandes Schweizerischer Buchdrucker, Zürich, 1951.

Goudy, F. W., «The Alphabet and Elements of Lettering», University of California Press, Berkeley/Los Angeles, 1942.

Johnston, E., «Schreibschrift, Zierschrift und angewandte Schrift», Leipzig, 1910, and «Writing and Illuminating and Lettering», London, 1906.

Kapr, A., «Schriftkunst», VEB Verlag der Kunst, Dresden, 1971.

Kindersley, D., «Optical Letter Spacing», in The Penrose Annual, The International Review of the Graphic Arts, Vol. 62, blz. 167-176, Lund Humphries Publishers Ltd., London, 1969.

Morison, S., «First Principles of Typography», Cambridge, 1936. «Letter Forms, Typographic and Scriptorial», London, 1968. «On Script Types», The Fleuron 4, London, 1925.

Ovink, G. W., «Legibility, Atmosphere-value and Forms of Printing Types», A. W. Sijthoff's Uitgeversmaatschappij n.v., Leiden, 1938.

Renner, P., «Die Kunst der Typographie», Bern, 1939.

Tracy, W., «Letters of Credit», a view of type design, London 1986

Treebus, K. F., «Het zetten van vreemde talen. Tips voor het juiste gebruik van accenten, hoofdletters, leestekens e.d. in het Deens, Duits, Engels, Esperanto, Fins, Frans, Fries, Hongaars, Iers, Indonesisch, Italiaans, Latijn, Noors, Pools, Portugees, Roemeens, Servo-Kroatisch, Slowaaks, Sloweens, Spaans, Tsjechisch, Turks, IJslands, Zuidafrikaans en Zweeds, alsmede beknopte regels voor het afbreken van woorden», Staatsuitgeverij, 's-Gravenhage, 1971.

Tschichold, J., «Die neue Typographie», Berlin, 1928. «Geschichte der Schrift in Bildern», Basel, 1946, and «Meisterbuch der Schrift«, Ravensburg, 1952.

Zapf, H., «About Alphabets. Some Marginal Notes on Type Design», New York, 1960. German edition: «Über Alphabete», Frankfurt am Main, 1960. «Manuale typographicum I-II», Frankfurt am Main, 1954/1968, and «Typographische Variationen», Frankfurt am Main, 1963.

2. Digital type

AD 90 Adobe Type1 font format / Adobe Systems Inc.,
ISBN 0-201-57044-0 / Addison-Wesley, Reading, 1991

AD 91 PostScript language reference manual / Adobe Systems Inc. - 2nd edition, ISBN 0-201-18127-4 / Addison-Wesley, Reading, 1991

AFII (Association for Font Information Interchange), «Registry Fon Showings», Fine Typography Latin, Los Alamitos, USA, 1991

AP 91 Apple Computer, The TrueType Book, Addison-Wesley, Reading 1991, and Microsoft Corp., TrueType Font Files Specification, Draf Release Version 1.00, Redmond, 1990

Bigelow, C. and Day, D., «Digital Typography», Scientific American, Vol 249, Number 2, August 1983, pp. 106-119.

Bigelow, C., «Font Design for Personal Workstations», BYTE magazine January 1985, pp. 255-270.

CG 91 Intellifont Scalable Typeface Format, Agfa-Compugraphi Division

Coueignoux, P., «Generation of Roman Printed Fonts» Ph. D. Thesis Massachusetts Institute of Technology, Cambridge, Massachusetts, Jun 1975.

IG URW, IG-Format, Interne Dokumentation, Hamburg, 1991

ISO/TC97/SC18/WG8, «Information Processing - Font and Character Information Interchange»; ANSI American National Standards Institute, 1430 Broadway, New York, NY 10018 c/o Bernadette St. John, Secretariat of ISO/TC97/SC18.

Karow, P., «Schrifttechnologie – Methoden und Werkzeuge», Springer Verlag, Heidelberg, August 1992.

Karow, P., «Typeface Statistics», URW Verlag, Hamburg, August 1993.

Knudson, D. R., «Digital Encoding of Newspaper Graphics», Massachusetts Institute of Technology, Cambridge, Report ESL-R-616, August 1975.

Knuth, D. E., «Metafont: A System for Alphabet Design», STAN-CS-79-762, Department of Computer Science, Stanford University, Stanford, California, September 1979. «TEX and Metafont» American Mathematical Society and Digital Press, Bedford, Massachusetts, 1979.

Naiman, A., «High-Quality Text for Raster Displays», Department of Computer Science, University of Toronto, January 1985.

Plass, P. and Stone, M., «Curve-Fitting with Piecewise Parametric Cubics», Imaging Sciences Laboratory, Xerox Palo Alto Research Centers, March 1983 (unpublished).

Ruggles, L., «Letterform Design Systems», STAN-CS-83-971, Department of Computer Science, Stanford University, Stanford, California, April 1983.

Späth, H., «Spline-Algorithmen zur Konstruktion glatter Kurven und Flächen», Oldenbourg Verlag München Wien, 1973.

SU 91 F3 Font Format Specification / Sun Microsystems, Inc.

Warnock, J. E., «The Display of Characters Using Gray Level Sample Arrays», Computer Graphics, Volume 14, Number 3, July 1980, pp. 302-307. Siggraph 1980 Proceedings

3. Journals

«Deutscher Drucker», Unabhängiges Fachmagazin für die Druckindustrie, Verlagsgesellschaft mbH & Co. KG, Stuttgart.

«Der Druckspiegel», Fachzeitschrift für deutsche und internationale Drucktechnik, Druckspiegel Verlagsgesellschaft mbH & Co., Heusenstamm.

«Page», Publizieren und Präsentieren mit dem Personal Computer, MACup-Verlag, Hamburg

«The Seybold Report on Publishing Systems», Seybold Publications Inc., Media (Pa).

«U&lc», Upper and Lower Case, The International Journal of Typographics, International Typeface Corp., New York.

«Visible Language», The Journal for Research on the Visual Media of Language Expression, Visible Language, Cleveland (Ohio).

Index

accented character 7
accents 7, 213, 214, 215, 216
aesthetic boxes 46
AFM, Adobe Font Metric 184
aid point 245
algorithmic 48
alignment point 433
AMS, American Mathematical Society 7
AMS Euler 7
anchor point 150
antiquied 14
application software 46
arch 106, 141
artwork 154
ascender 8, 9
ascender height 157
ASCII 59
ASCII form 144
ATM, Adobe Type Manager 151, 179
ATYPI, Association Typographique International 5, 199
auxiliary line 123
auxiliary points 421
axis of bowl 9
axis, X-, Y- 430
back classes 389
bar 103, 106, 141, 142
bar serifs 141
baseline 6, 9
base overhang 9, 136
BE format 91
beam, electron = 22
beam, light = 23, 35
bearing, left side = 13, 45, 421
bearing, right side = 13, 45, 421
bearing, unequal = 111
beginning point 65
Bézier control point 74
Bézier curve 74, 143, 180, 186
Bézier format 255
Bézier function 70, 72, 94
Bézier knot 74
BI format 91
binary address 19
binary format 106
bit 54
bitmap 57, 69
bitmap editor 98
Blackletter 5, 204
black encodeing 81
bodysize 17, 49
boldness 135
borderline 171
bounding box 12
bow 9, 106, 141
BS format 91, 100, 106, 109
bytemap 26, 27, 57, 82
byte sequence 169
CAD, computer aided design 30
capheight 9, 49, 157
cap character 158
capital letters 6, 136
caps 6
Cartesian coordinate system 244

Cartesian geometry 12
channeling 118
channel processing 109
character address table 45
character contour 161
character elements 148
character feature 159
character geometry 153
character image 161
character outline 166
character pair 389, 390
character parts 430, 431
chord height 84
circle 50, 70, 423
circle fitting 94
circle parameter 97
clipping 97
closed curve 94
closed outline 185
closed vector 67
clothoid 9
clustering method 389
CN format 91
code 57
coding 45
coefficient 97
color coding 63
compression of data 85
conic format 255
contour index 327
contour, inner = 149
contour, outer = 149
contour point 185
control point 72
control signal 188
coordinate origin 430
core-font 98
corner point 424, 425, 432
correction, interactive = 132
correction, manual 132
correction value 154
correction value 389, 390
Coueignoux format 79
counter 8, 9, 141
counter punch 8
CPU 189
cross-hair 427, 428
crossbar 9
CRT (cathode ray tube) 1
CRT machine 30, 36
Cursive 5
curvature variation 97
curve 58, 70, 85
curve bar 141
curve extreme 112, 421
curve fit calculation 94
curve point 151, 245
curve run 424
curve stem 8, 103, 141
curve switches 111
curved line 70
curved path 96, 119
curved stroke 119, 129
damping term 97
data base 192, 193
data structure 235
DDA, Digital Differential Analyzer 97
delimiter 157
delta 141
descender 8, 9
descriptive elements 103, 106, 139
design integrity 157, 179
desktop publishing 97
DI format 91, 97, 118, 120
diagonal 9
Didot point 17
digital formats 19
digital machine format 3
digital storgae 45

digital type 117
digital units 13
digitally stored type 1
digitization 4, 72
digitized point 94
digitizing 430
digitizing character 431
digitizing procedure 421, 430
digitizing rules 427
dingbats 217, 218
display 5
display application 160
display format 97
display, graphic = 70
display, graphic visual = 20
display, halftone = 26, 27
display, raster scan = 22
Display Roman 202
display screen 139
distance control 175
dot matrix printer 1
dots per inch 15
drafting devices 139
drawing machine 70, 185
dropout 141, 158, 167
DTP 101
ductus 202
ductus change 119
DWU, Design Window Unit 154
ear 9
eccentires 204
Egyptian slab 201
electronic communcation 2
electronic noise 101
em raster 89
em square 192
ending point 65
engraver 39
engraving stylus 430
excesse 12
exponents 6
exposing window 192
extrapolate 2
extreme 106, 141
F3 font 181
FAIS font 154
FAIS format 153, 156
Fancy 5
finest raster 93
fixed accented character 45, 46
fixed accents 193
flame 39
flat curve 128
flatbed drawing 40, 42
flatbed drawing machine 186
flatbed plotter 39, 42
floating accents 45, 209
flutes 131
foil cutting machine 43
font control data 174
font data 155
font design 143
font downloading 160
font format 236
font guide line 280
font header 183
font information 276
font program 143
font reference table 280
font scaling 101
format definition block 276
fractals 158
fraction 6
frame 2, 54
French curve 74
front classes 389
gaps 167, 243
Gothic 195, 202
graphic 53, 54, 203

graphic display terminal 25
graphical user interface 179
grid 16
grid fitting 119, 126
GS format 91
H-height 17
hairline 9
halfbyte 24, 26, 27
halftone 8
half bitting 137
hand-digitization 89, 94
hand-digitized 90
hand-digitizing 435
hard scanner 100
hard scanning 100, 117
hierarchy 93
high resolution 18, 70
hints 148
hint command 151
host computer 193
icon 140
II format 91, 104
IK character data 282
IK format 104
IK interpolation 94
IK interpolation 186
IKARUS concept 190
IKARUS directory 281
IKARUS format 74
IKARUS point 76
IKARUS system 103
imagesetter 34
image information 235, 328
inclination 106
indices 6
individual access 55
inferiors 6
information block 277
ink traps 8, 133, 134
inner edge 49, 50
instruction 139
instruction data 284
integer operation 103
integral sign 7
Intellifont 100, 153
intelligence 103, 161
intelligent fontscaling 98, 103, 153
intelligent outline 106
intelligent outline font 179
intelligent sizing 139
interpolation function 178
ISAM, index sequential access method 54
italic, slant 6
italic type 130
ITC, International Typeface Corporation 6
ITC Dingbats 7
ITC Zapf Dingbats 105
Kanji 191, 195
Kanji fonts 236
Kanji type characater 194
»Kerning on the Fly» 48
kerning 156, 389
kerning table 85, 156, 160, 190, 401
kerning values 46
keyboard layout 190
kissing table 190
knife 39
knot 25
knot 25, 77, 94
laserwriter 143
laser beam 39
laser composer 39
laser film recorder 37
laser typewriter 21
laser vaporization 38
Latin fonts 236
Latin letter 50

leading 192
letter height 185
letter outline 90
letter pairs 46
letter spacing 389, 390
letter touching 97
LIFO structure 169
ligature 14, 45, 46
line border 7
line element 7, 54
line element 54
line graphics 2, 7, 243
line linking 167
line repeat 82
line separation 93
line spacing 192
link 9
LINUS 92
logo 2, 7
logo 2, 7, 50
logotype 421
long kerning table 47, 389
long table 390
lowercase 6, 206
lowercase character 158
lowercase height 6
low resolution 18
machine format 4, 89
magnetic storage media 5
margin justification 97
master grid 112
master resolution 110
mathematical formulae 7
mathematical formula table 48
maximum length 390
mean value 12
megabit 115
memory capacity 57
menu field 432
Metafont 58
Metafont concept 80
Metafont typeface 80
microelectronic 89
microfilming 35, 39
microfilm device30
miniscules 6
mirrorimage 85
mistake of recognition 10
Modern face 201
Modern Roman 201
module, electronic = 45
monospace 15
mutt & nutt 12
NC 39
NC machine (numeric controlled) 1, 97
neighbouring straights 245
nibble 83, 85
NIMBUS 97, 181
non-proportional 15
numerals 6, 207
numeric value 150
Old face 200
Old style 200
»on the fly» 97
optimal images 139
original letter 185
original point 94
ornamentation element 7
outer edge 49, 50
outline description 98
outline font format 100
outline geometry 155
outline representation 164
overhang 8
overhang of curves 123
overhang zone 147
overlapping 47, 49, 389
overlapping table 402

PA program 013
parabola section 165
part serifs 430
PC 31
peak 106
peak adjustment 113
pen 39
phototypesetting 30
Pi fonts 7
pice point 17
pixel 16, 89, 136
pixel border 171
pixel center 171
pixel gap 158
plane figure 243
plotter 41
point, colinear = 160
point coordinate 178
point, key = 158
point, skeletal = 158, 159
pointsize 6, 135
polygon path 63
polynom fit 94
polynomial 72
PostScript format 143
PostScript languange 143
PostScript Type1 144, 181
preference 159
printed letter 5
printer, dot matrix = 31, 38, 97
printer, electrostatic = 30, 33
printer, laser = 30, 34, 135, 139
printer, matrix = 30, 32
printing device 15
printing plate 1
proportional type display 3
punctuation marks 6, 124, 208
Purdy format 74
pure rasterization 111
QQ format 91
quadratic spline format 255
quarterbyte 26, 27
RAM 189
random 54
random access 54
raster 85
raster display terminal 24
raster dot 123
raster form 117
raster pattern 193
raster per em 190
raster point 23, 93, 123
raster size 124
raster skip 118
raster step 12
rasterimage 171
rastering work 196
rastersize 119
reference point 12, 173, 175
resolution 23, 24
resolution funnel 115
resolution, high = 38
resolution, low = 38, 135
RIP 143
RIP, raster image processor 97, 98
Roman 5
Roman character 8
Roman, upright 6
root sign 7
rotation 167
rounding 172
rounding function 178
router 39
run length 57, 63, 82, 84
run length code 187
run length format 98
sandjet 39
Sans serif 5, 9, 202
SC format 91, 117

scaling 103, 139
scaling program 140
scaling techniques 139
scanline 63, 125, 126
scan control 168
scanline data 92
scanline skip 129
scanline step 123
scanner 3
scanning 58
Script 5
scripts 203
sector kerning 156
segmentation 58
sensor key 427
serif 8, 122, 130, 141, 431
serif adjustment 112
Serif Antique 201
serif length 131
set with 135
sexy setting 47
shallow curve 152, 159
shoulder 9
SIGNUS system 97
sign change 85
skewing 167
slab serif 201
slant 141
slight slopes 133
small caps 6
small stroke 8
smooth 243
SN format 91
soft-scanned line 120
soft scanning 100, 117, 118, 187
sources 28
special characters 208, 209, 210
spiral 70
spiral segment 74
spline 9, 70, 76
spline, circle = 74
spline, conic = 70, 74
spline function 72, 92, 97
spline interpolation 94, 186
spline method 77
spline, quadratic = 70
spot 141
spur 9
square 12
stair stepping 159
start point 151, 245, 432
stem 9, 103, 106, 141, 142
stem adjustment 109, 110
stem block 326
stem control 171
stem, diagonal = 122, 130
stem, straight = 127
step structure 121
storage display terminal 26
storage requirement 151
straight 74
straight contour 425
straight line 67
straight stem 8, 107, 432
stress 9
stroke intersection 195
stroke thickness 49, 127
stroke width 126
subscript 6, 46
subscript character 158
superiors 6
superscript 6, 46
swash 9
swash character 46
swash element 7
switches 139
symbol 2, 50, 243

symbol set 105
symmetrizing 109
T1 format 100
tails 134
tangent 9, 76
tangent calculation 95
tangent point 70, 424, 425, 426, 432
tangential 9
tangential transition 70, 74
target grid 181
target raster 93
tension 141
terminal 201
text 5
total width 45
touching 389
touching table 46, 85, 193, 401
traditional 389
transformation matrix 181
Transitional 200
TrueType 100, 161
TrueType table 162
Type1 144
typefaces 52
type character 243
type format 85
type information 57
typeface layout 194
typeface structure 54
typesetter, laser = 30
typesetter-like processor 29
typesetting machine 1, 3, 8, 30, 38, 97, 139
typesetting, monospace = 32
typesetting speed 39
typographic character 2
unitizing 109
upper case 205
upward 83
variables 178
VC format 84, 91
VDT 25, 46
VDT, alphanumeric 3
VDT (video display terminal) 22
VDT, visual display device 1
VE format 91
vector 20, 57
vector format 83, 85
vector, freedom = 170
vector, open = 65
vector, projection = 170
vector repeat 85
vector/circle outline 100
Venetian serif 199
vertical raster 16
vertical stright 103
vignette 2
vinyl cutting machine 43
waterjet 39
wedge 8
weight 135, 141
width table 45, 190
writing machine 3
WYSIWYG 115
x-height 9, 49, 157
zero point 12

Production:
Cover: Calligraphy by Jovica Veljović
Translation: Bill Horton
Correction: Jochen Lau
Layout: Gerd Kretzschmar
Typesetting: QuarkXPress
Graphic system: Illustrator, FreeHand, Canvas

Typeface:
Name: Latino, URW Grotesk
Designer: Hermann Zapf
Typesetter: Compugraphic 9000

Printing: Druckerei Zechner, Speyer
Binding: Buchbinderei Schäffer, Grünstadt